Combining Forecasts under Structural Breaks Using Graphical LASSO

Tae-Hwy Lee* and Ekaterina Seregina†

March 14, 2025

Abstract

In this paper we develop a novel method of combining many forecasts based on Graphical LASSO. We represent forecast errors from different forecasters as a network of interacting entities and generalize network inference in the presence of common factor structure and structural breaks. First, we note that forecasters often use common information and hence make common errors, which makes the forecast errors exhibit common factor structures. We separate common forecast errors from the idiosyncratic errors and exploit sparsity of the precision matrix of the latter. Second, since the network of experts changes over time as a response to unstable environments, we propose Regime-Dependent Factor Graphical LASSO (RD-FGL) that allows factor loadings and idiosyncratic precision matrix to be regime-dependent. The empirical applications to forecasting macroeconomic series using the data of the European Central Bank's Survey of Professional Forecasters and Federal Reserve Economic Data monthly database demonstrate superior performance of a combined forecast using RD-FGL.

Keywords: Common Forecast Errors, Regime Dependent Forecast Combination, Sparse Precision Matrix of Idiosyncratic Errors, Structural Breaks.

JEL Classifications: C13, C38, C55

We are grateful to Co-Editor George Kapetanios, an anonymous Associate Editor, and two anonymous reviewers for their constructive comments and suggestions. We thank the participants of seminars at Duke, North Carolina State University, and the Midwest Econometrics Group (MEG), especially Bin Chen, Mehmet Caner, Dennis Pelletier, Ilze Kalnina, and Anna Bykhovskaya, for their comments and suggestions.

The numerical results presented in the manuscript were reproduced by the Editor-in-Chief on 30 April 2025.

*Department of Economics, University of California Riverside. Email: taelee@ucr.edu.

†Department of Economics, Colby College. Email: eseregin@colby.edu.

1 Introduction

A search for the best forecast combination has been an important on-going research question in economics. Clemen (1989) pointed out that combining forecasts is "practical, economical and useful. Many empirical tests have demonstrated the value of composite forecasting. We no longer need to justify that methodology". However, as demonstrated by Diebold and Shin (2019), there are still some unresolved issues. Despite the findings based on the theoretical grounds, equal-weighted forecasts have proved surprisingly difficult to beat. Many methodologies that seek for the best forecast combination use equal weights as a benchmark: for instance, Diebold and Shin (2019) develop "partially egalitarian LASSO".

The success of equal weights is partly due to the fact that the forecasters use the same set of public information to make forecasts, hence, they tend to make common errors. For example, in the European Central Bank's Survey of Professional Forecasters (ECB SPF) of Euro-area real GDP growth, the forecasters tend to *jointly* understate or overstate GDP growth. This stylized fact is illustrated in Figure 1 that shows quarterly forecasts of Euro-area real GDP growth produced by the ECB SPF from 1999Q3 to 2024Q1. As described in Diebold and Shin (2019), the ECB SPF forecasts are solicited for two quarters ahead of the latest available outcome. The forecasts are concentrated either below or above the actual series. Hence, forecasters tend to jointly understate or overstate GDP growth. Similar pattern is observed for Federal Reserve Economic Data monthly database (FRED-MD, McCracken and Ng (2016)) that includes data on 126 macroeconomic time series. Figure 2 plots two-step ahead forecasts of the average growth rate of industrial production (INDPROD) based on individual monthly indicators. Similarly to Figure 1, the forecasts tend to jointly understate or overstate INDPROD growth rate. Therefore, we stipulate that the forecast errors include

common and idiosyncratic components, which allows the forecast errors to move together due to the common error component. Our paper provides a simple framework to learn from analysing forecast errors: we separate unique errors from the common errors to improve the accuracy of the combined forecast.

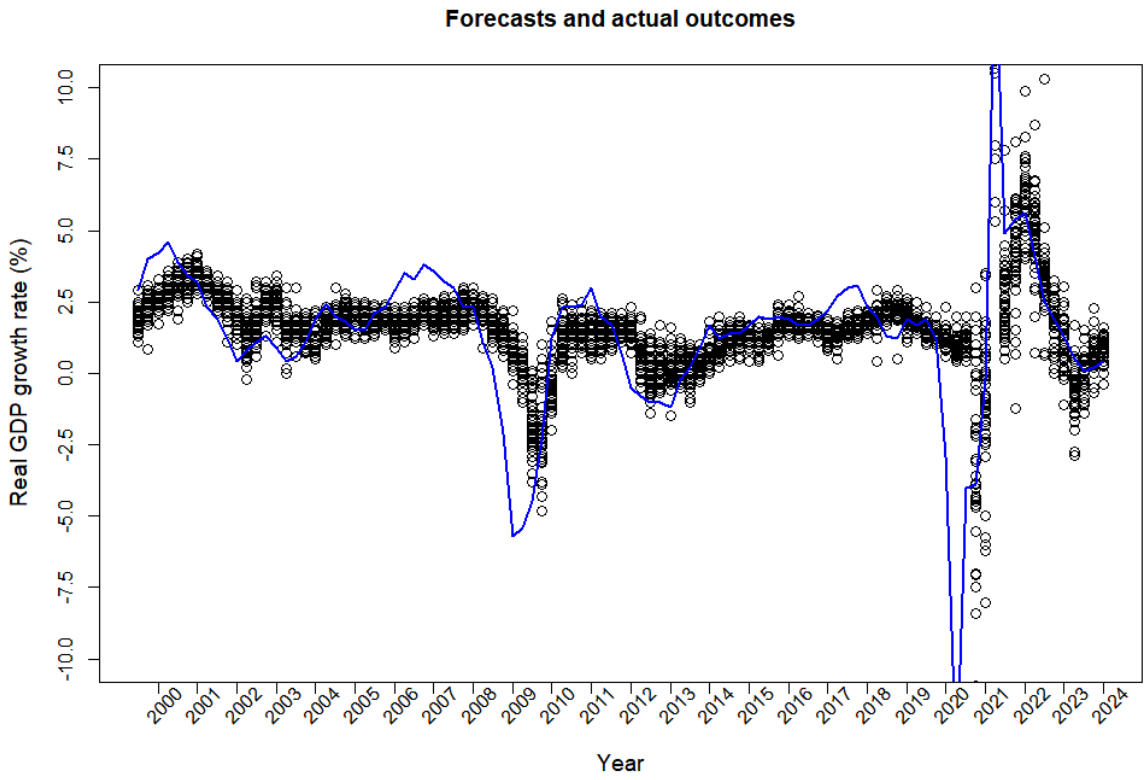

Figure 1: **ECB SPF on Real GDP Growth**. Each circle denotes the forecast of each professional forecaster in the SPF for the quarterly 2-quarters-ahead forecasts of Euro-area real GDP growth, year-on-year percentage change. Actual series is the blue line. *Source: European Central Bank*.

Our paper aims at improving prediction of the target series using information from multiple available known (ECB-SPF) or estimated (FRED-MD) forecasts. This requires estimating inverse covariance (precision) matrix for the optimal combination weight (Bates and Granger (1969)) which is challenging in high dimensions. Building on network theory and

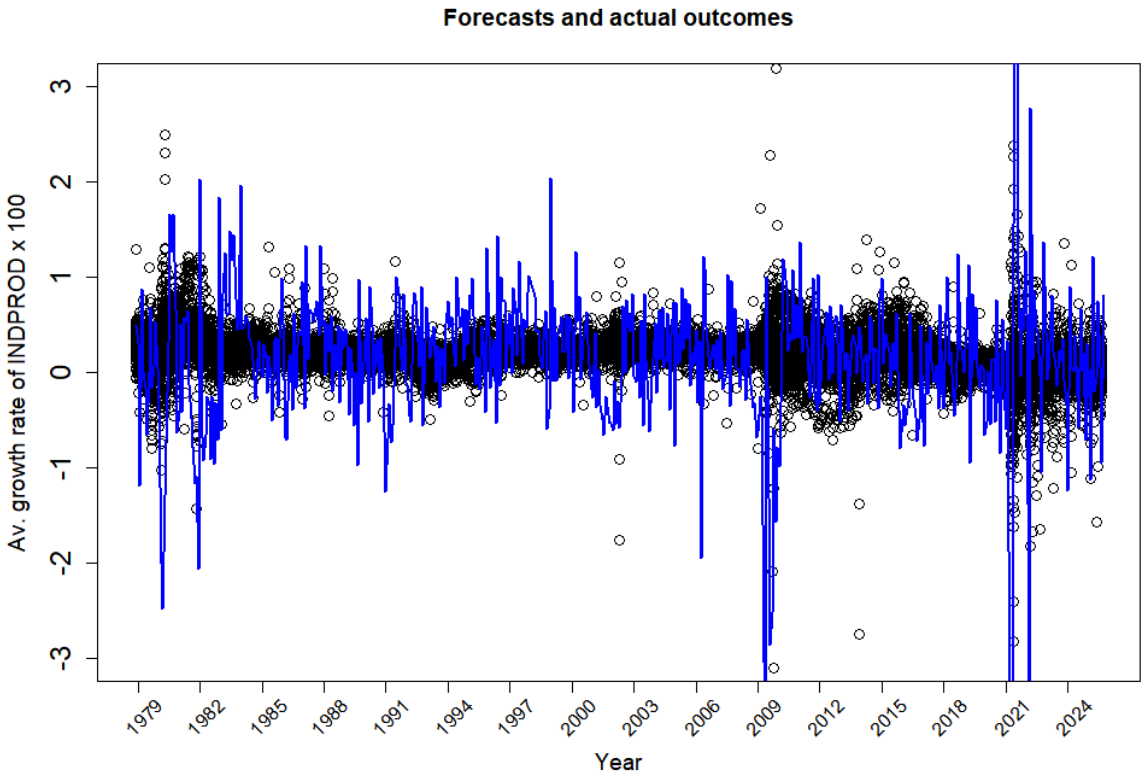

Figure 2: **FRED-MD on industrial production (INDPROD)**. Each circle denotes two-step ahead forecast of the average growth rate of INDPROD based on individual monthly indicators (125 forecasting models per period). Actual series is the blue line.

graphical models, we provide a novel way to estimate precision matrix that achieves stability and sparsity and can handle structural breaks.

Precision matrix represents a network of interacting entities, such as corporations or genes. When the data is Gaussian, the sparsity in the precision matrix encodes the conditional independence graph - two variables are conditionally independent given the rest if and only if the entry corresponding to these variables in the precision matrix is equal to zero. Graphical models are a powerful tool to directly estimate precision matrix, avoiding the step of obtaining an estimator of covariance matrix to be inverted. Prominent examples of graphical models include Graphical LASSO (GL, Friedman et al. (2008)) and nodewise

regression (Meinshausen and Bühlmann (2006)). Despite using different strategies for estimating precision matrix, all graphical models assume that precision matrix is sparse: many entries of precision matrix are zero, which is a necessary condition to consistently estimate inverse covariance. Our paper demonstrates that such assumption contradicts the stylized fact that experts tend to make common errors and hence the forecast errors move together through common factors. Lee and Seregina (2024) show that graphical models fail to recover the entries of a nonsparse precision matrix under the factor structure and propose Factor Graphical LASSO (FGL) that combines the benefits of graphical models and factor models.

At the same time, the network of experts changes over time, that is, the relationships between forecasts produced by different experts or models can change either smoothly or abruptly (e.g., as a response to an unexpected policy shock, or in the times of economic downturns). Such changes give rise to different regimes and it is important to account for changes in optimal forecast combination weights induced by structural breaks (Corradi and Swanson (2014)). The co-movement of forecasters in Figures 1-2 changes over time, and is especially strong in certain periods, such as during crises. It is reasonable to expect that during such periods the network of forecasters will change. This change could come from two sources: (i) the change in the co-movement between forecasters, and (ii) the change in idiosyncratic behavior of forecasters in the periods of increased uncertainty. This paper augments Lee and Seregina (2024) and develops a framework to generalize network inference in the presence of structural breaks. As a first extension, to capture the change that comes from common movements, we model structural changes in factor loadings. As a second extension, to capture the change that comes from idiosyncratic movements, we model structural changes in the precision matrix of the idiosyncratic component after removing common factors. To model structural changes in factor loadings, we use a kernel that weighs recent observations

more than distant ones, and we give weight 1 to post-break observations, and weight γ to pre-break observations. To model structural changes in idiosyncratic precision matrix, we augment GL with a penalty that enforces temporal constancy and controls the strengths of resemblance between two neighboring precision estimators. We consider two definitions of regimes: a regime defined by a time break, and a regime determined by a state variable. The time and location of a break is allowed to be unknown and is estimated using the framework used in Bai et al. (2020) and Bai (2010). We estimate regime-dependent precision matrix for forecast combination using both pre- and post-break data when forecast errors are driven by common factors. We call the proposed algorithm *Regime-Dependent Factor Graphical LASSO* (RD-FGL) and develop its scalable implementation using the Alternating Direction Method of Multipliers (ADMM).

Our paper makes several contributions. First, we allow the forecast errors to be highly correlated due to the common component which is motivated by the stylized fact that the forecasters tend to jointly understate or overstate the predicted series of interest. Second, to tackle changing relationships between forecasts produced by different experts or models as a response to unstable environments, we develop a framework to generalize network inference in the presence of structural breaks. We propose Regime-Dependent Factor Graphical LASSO (RD-FGL) that models structural changes in factor loadings and idiosyncratic precision matrix. We develop scalable implementation of RD-FGL using ADMM to estimate regime-dependent forecast combination weights. Third, two empirical applications to forecasting macroeconomic series using the data of the ECB SPF and FRED-MD shows that incorporating (i) factor structure in the forecast errors together with (ii) sparsity in the precision matrix of the idiosyncratic components and (iii) regime-dependent combination weights improves the performance of a combined forecast over forecast combinations using

equal weights.

The paper is structured as follows. Section 2 studies the approximate factor model for the forecast errors, reviews FGL and contains theoretical results on the consistency of the FGL estimator for forecast combinations. Section 3 introduces Regime-Dependent graphical model and discusses its implementation using ADMM. Section 4 studies an empirical application to combining ECB SPF forecasts. Section 5 presents another empirical application to FRED-MD. Section 6 concludes. Section 7 provides a link to online Github repository for reproducing the results in the paper.

Notation. For the convenience of the reader, we summarize the notation to be used throughout the paper. Let $\mathcal{S}_p$ denote the set of all $p \times p$ symmetric matrices. For any matrix $\mathbf{C}$, its (i,j)-th element is denoted as c_{ij}. Given a vector $\mathbf{u} \in \mathbb{R}^d$ and a parameter $a \in [1,\infty)$, let $\|\mathbf{u}\|_a$ denote ℓ_a-norm. Given a matrix $\mathbf{U} \in \mathcal{S}_p$, let $\lambda_{\max}(\mathbf{U}) \equiv \lambda_1(\mathbf{U}) \geq \lambda_2(\mathbf{U}) \geq \ldots \geq \lambda_{\min}(\mathbf{U}) \equiv \lambda_p(\mathbf{U})$ be the eigenvalues of $\mathbf{U}$. Given a matrix $\mathbf{U} \in \mathbb{R}^{p\times p}$ and parameters $a, b \in [1,\infty)$, let $|||\mathbf{U}|||_{a,b} \equiv \max_{\|\mathbf{y}\|_a=1} \|\mathbf{U}\mathbf{y}\|_b$ denote the induced matrix-operator norm. The special cases are $|||\mathbf{U}|||_1 \equiv \max_{1\leq j\leq p} \sum_{i=1}^p |u_{ij}|$ for the ℓ_1/ℓ_1-operator norm; the operator norm (ℓ_2-matrix norm) $|||\mathbf{U}|||_2^2 \equiv \lambda_{\max}(\mathbf{U}\mathbf{U}')$ is equal to the maximal singular value of $\mathbf{U}$. Finally, $\|\mathbf{U}\|_{\max} \equiv \max_{i,j}|u_{ij}|$ denotes the element-wise maximum.

2 Approximate Factor Models for Forecast Errors

We are interested in finding the combination of forecasts which yields the best out-of-sample performance in terms of the mean-squared forecast error. Motivated by the stylized facts discussed in the introduction and depicted in Figures 1-2, we claim that the forecasters use the same set of public information to make forecasts and, hence, they tend to make

common errors. Therefore, we can model the tendency of the forecast errors to move together via factor decomposition. Figures F.1-F.2 in Supplemental Appendix F provide similar stylized facts for inflation and unemployment rate from the ECB SPF dataset, and Figures F.3-F.5 provide illustrations for consumer price index, personal consumer expenditures, and civilian unemployment rate from the FRED-MD dataset.

Suppose we have p competing forecasts of the univariate series y_t, $t = 1, \ldots, T$ and $\widetilde{\mathbf{e}}_t = (\widetilde{e}_{1t}, \ldots, \widetilde{e}_{pt})' \sim \mathcal{N}(\mathbf{m}_t, \mathbf{\Sigma})$ is a $p \times 1$ vector of forecast errors. Note that we allow bias $\mathbf{m}_t$ in the forecast errors, which is allowed to be time-varying. Assume that the generating process for the forecast errors follows a q-factor model: $\widetilde{\mathbf{e}}_t = \mathbf{m}_t + \mathbf{B}\mathbf{f}_t + \boldsymbol{\varepsilon}_t$, where $\mathbf{f}_t = (f_{1t}, \ldots, f_{qt})'$ are the common factors of the forecast errors for p models, $\mathbf{B}$ is a $p \times q$ matrix of factor loadings, and $\boldsymbol{\varepsilon}_t$ is the idiosyncratic component that cannot be explained by the common factors. Define demeaned forecast errors as $\mathbf{e}_t \equiv \widetilde{\mathbf{e}}_t - \mathbf{m}_t$ such that:

$$\underbrace{\mathbf{e}_t}_{p\times 1} = \mathbf{B} \underbrace{\mathbf{f}_t}_{q\times 1} + \boldsymbol{\varepsilon}_t, \quad t = 1, \ldots, T, \tag{2.1}$$

Unobservable factors, $\mathbf{f}_t$, and loadings, $\mathbf{B}$, are estimated by the principal component analysis (PCA), studied in Bai and Ng (2002); Stock and Watson (2002), and the estimators are denoted as $\widehat{\mathbf{f}}_t$ and $\widehat{\mathbf{B}}$.

We use the following notations: $\mathbb{E}[\boldsymbol{\varepsilon}_t\boldsymbol{\varepsilon}_t'] = \mathbf{\Sigma}_\varepsilon$, $\mathbb{E}[\mathbf{f}_t\mathbf{f}_t'] = \mathbf{\Sigma}_f$, and $\mathbb{E}[\mathbf{e}_t\mathbf{e}_t'] = \mathbf{\Sigma} = \mathbf{B}\mathbf{\Sigma}_f\mathbf{B}' + \mathbf{\Sigma}_\varepsilon$. Let $\mathbf{\Theta} = \mathbf{\Sigma}^{-1}$, $\mathbf{\Theta}_\varepsilon = \mathbf{\Sigma}_\varepsilon^{-1}$ and $\mathbf{\Theta}_f = \mathbf{\Sigma}_f^{-1}$ be the precision matrices of forecast errors, idiosyncratic and common components respectively.

Given a sample of the estimated residuals $\{\widehat{\boldsymbol{\varepsilon}}_t = \mathbf{e}_t - \widehat{\mathbf{B}}\widehat{\mathbf{f}}_t\}_{t=1}^T$ and the estimated factors $\{\widehat{\mathbf{f}}_t\}_{t=1}^T$, let $\widehat{\mathbf{\Sigma}}_\varepsilon = (1/T)\sum_{t=1}^T \widehat{\boldsymbol{\varepsilon}}_t\widehat{\boldsymbol{\varepsilon}}_t'$ and $\widehat{\mathbf{\Sigma}}_f = (1/T)\sum_{t=1}^T \widehat{\mathbf{f}}_t\widehat{\mathbf{f}}_t'$ be the sample counterparts of the covariance matrices.

Moving forward to the forecast combination exercise, suppose we have p competing forecasts, $\widehat{\mathbf{y}}_t = (\hat{y}_{1,t}, \ldots, \hat{y}_{p,t})'$, of the variable y_t, $t = 1, \ldots, T$. The forecast combination is defined as $\widehat{y}_t^c = \mathbf{w}'\widehat{\mathbf{y}}_t$, where $\mathbf{w}$ is a $p \times 1$ vector of weights. Define a measure of risk $\text{MSFE}(\mathbf{w}, \boldsymbol{\Sigma}) = \mathbf{w}'\boldsymbol{\Sigma}\mathbf{w}$. As shown in Bates and Granger (1969), the *optimal* forecast combination minimizes the MSFE of the combined forecast error:

$$\min_{\mathbf{w}} \text{MSFE} = \min_{\mathbf{w}} \mathbb{E}\left[\mathbf{w}'\mathbf{e}_t\mathbf{e}_t'\mathbf{w}\right] = \min_{\mathbf{w}} \mathbf{w}'\boldsymbol{\Sigma}\mathbf{w}, \text{ s.t. } \mathbf{w}'\boldsymbol{\iota}_p = 1, \tag{2.2}$$

where $\boldsymbol{\iota}_p$ is a $p \times 1$ vector of ones. The solution to (2.2) yields a $p \times 1$ vector of the optimal forecast combination weights[1]:

$$\mathbf{w} = \frac{\boldsymbol{\Theta}\boldsymbol{\iota}_p}{\boldsymbol{\iota}_p'\boldsymbol{\Theta}\boldsymbol{\iota}_p}. \tag{2.3}$$

If the true precision matrix is known, the equation (2.3) guarantees to yield the optimal forecast combination. In reality, one has to estimate $\boldsymbol{\Theta}$. As pointed out by Smith and Wallis (2009) and Claeskens et al. (2016), when the estimation uncertainty of the weights is taken into account, there is no guarantee that the "optimal" forecast combination will be better than the equal weights or even improve the individual forecasts. Define $a = \boldsymbol{\iota}_p'\boldsymbol{\Theta}\boldsymbol{\iota}_p/p$ and $\widehat{a} = \boldsymbol{\iota}_p'\widehat{\boldsymbol{\Theta}}\boldsymbol{\iota}_p/p$. We can write $\left|\frac{\text{MSFE}(\widehat{\mathbf{w}},\widehat{\boldsymbol{\Sigma}})}{\text{MSFE}(\mathbf{w},\boldsymbol{\Sigma})} - 1\right| = \left|\frac{\hat{a}^{-1}}{a^{-1}} - 1\right| = \frac{|a-\hat{a}|}{|\hat{a}|}$ and $\|\widehat{\mathbf{w}} - \mathbf{w}\|_1 = \Big[(a\widehat{\boldsymbol{\Theta}}\boldsymbol{\iota}_p) - (a\boldsymbol{\Theta}\boldsymbol{\iota}_p) + (a\boldsymbol{\Theta}\boldsymbol{\iota}_p) - (\hat{a}\boldsymbol{\Theta}\boldsymbol{\iota}_p)\Big] / \Big(p \cdot \hat{a}a\Big)$. Therefore, in order to control the estimation uncertainty in the MSFE and combination weights, one needs to obtain a consistent estimator of the precision matrix $\boldsymbol{\Theta}$.

Graphical models such as Graphical Lasso (Friedman et al. (2008), Appendix A) are natural candidates for estimating a high-dimensional precision metrix. Lee and Seregina

[1]Our paper focuses on unconditional combination weights as in Bates and Granger (1969). Gibbs and Vasnev (2024) consider forecast combination weights conditioned on the information set available at time t.

(2024) introduce Factor Graphical LASSO (FGL) that bridges graphical models and factor models to estimate precision matrix in the presence of common factors (Appendix B). First, they use the Weighted Graphical Lasso penalty:

$$\widehat{\boldsymbol{\Theta}}_{\varepsilon,\tau} = \arg\min_{\boldsymbol{\Theta}_\varepsilon = \boldsymbol{\Theta}_\varepsilon'} \operatorname{tr}(\mathbf{W}_\varepsilon \boldsymbol{\Theta}_\varepsilon) - \log\det(\boldsymbol{\Theta}_\varepsilon) + \tau \sum_{i \neq j} \widehat{\gamma}_{\varepsilon,ii}\widehat{\gamma}_{\varepsilon,jj}|\theta_{\varepsilon,ij}|, \tag{2.4}$$

initialized with $\mathbf{W}_\varepsilon = \widehat{\boldsymbol{\Sigma}}_\varepsilon + \tau\mathbf{I}$, where $\widehat{\gamma}_{\varepsilon,ii}$ is the (i,i)-th element of $\widehat{\boldsymbol{\Gamma}}_\varepsilon^2 \equiv \operatorname{diag}(\mathbf{W}_\varepsilon)$. The subscript τ in $\widehat{\boldsymbol{\Theta}}_{\varepsilon,\tau}$ means that the solution of the optimization problem in (2.4) will depend upon the choice of the tuning parameter τ. In order to simplify notation, we will omit the subscript τ.

Second, they use Sherman-Morrison-Woodbury formula to estimate the precision of forecast errors:

$$\widehat{\boldsymbol{\Theta}} = \widehat{\boldsymbol{\Theta}}_\varepsilon - \widehat{\boldsymbol{\Theta}}_\varepsilon\widehat{\mathbf{B}}[\widehat{\boldsymbol{\Theta}}_f + \widehat{\mathbf{B}}'\widehat{\boldsymbol{\Theta}}_\varepsilon\widehat{\mathbf{B}}]^{-1}\widehat{\mathbf{B}}'\widehat{\boldsymbol{\Theta}}_\varepsilon. \tag{2.5}$$

$\widehat{\boldsymbol{\Theta}}$ is then used to estimate the forecast combination weights $\widehat{\mathbf{w}} = \widehat{\boldsymbol{\Theta}}\boldsymbol{\iota}_p/\boldsymbol{\iota}_p'\widehat{\boldsymbol{\Theta}}\boldsymbol{\iota}_p$. This approach allows to extract the benefits of modelling common movements in forecast errors, captured by a factor model, and the benefits of using many competing forecasting models that give rise to a high-dimensional precision matrix, captured by a graphical model.[2]

The asymptotic properties of FGL were studied in Lee and Seregina (2024), below we briefly list the main assumptions and extend their result to the context of optimal forecast combination.

[2]Note that we allow negative weights. As pointed out by Radchenko et al. (2023), negative weights emerge when highly correlated forecasts with similar variances are combined. In this situation, the estimated weights have large variances, and trimming is usually applied to reduce the variance of the estimated weights. This problem arises due to unstable inverse covariance matrix in the case of highly correlated forecast errors. Our paper explicitly models the high correlations in forecast errors using a factor model and uses the estimates to get total precision matrix. Sections 2 and 3 prove that the resulting precision matrix estimator and forecast combination weights are consistent; Supplemental Appendix E verifies consistency by simulations and shows that the estimators are robust to various data generating processes.

Let $A \in \mathcal{S}_p$. Define the following set for $j = 1, \ldots, p$:

$$D_j(A) \equiv \{i : A_{ij} \neq 0,\ i \neq j\}, \quad d_j(A) \equiv \text{card}(D_j(A)), \quad d(A) \equiv \max_{j=1,\ldots,p} d_j(A), \tag{2.6}$$

where $d_j(A)$ is the number of edges adjacent to the vertex j (i.e., the *degree* of vertex j), and $d(A)$ measures the maximum vertex degree. Define $S(A) \equiv \bigcup_{j=1}^{p} D_j(A)$ to be the overall off-diagonal sparsity pattern, and $s(A) \equiv \sum_{j=1}^{p} d_j(A)$ is the overall number of edges contained in the graph.

(A.1) (Spiked covariance model) (i) As $p \to \infty$, $\lambda_1(\mathbf{\Sigma}) > \lambda_2(\mathbf{\Sigma}) > \ldots > \lambda_q(\mathbf{\Sigma}) \gg \lambda_{q+1}(\mathbf{\Sigma}) \geq \ldots \geq \lambda_p(\mathbf{\Sigma}) > 0$, where $\lambda_j(\mathbf{\Sigma}) = \mathcal{O}(p)$ for $j \leq q$, while the non-spiked eigenvalues are bounded, that is, $c_0 \leq \lambda_j(\mathbf{\Sigma}) \leq C_0$, $j > q$ for constants $c_0, C_0 > 0$. (ii) $\boldsymbol{\iota}_p' \mathbf{\Theta} \boldsymbol{\iota}_p / p \geq c$, where c is a positive constant.

(A.2) (Pervasive factors) There exists a positive definite $q \times q$ matrix $\breve{\mathbf{B}}$ such that $\left|\left|\left| p^{-1}\mathbf{B}'\mathbf{B} - \breve{\mathbf{B}} \right|\right|\right|_2 \to 0$ and $\lambda_{\min}(\breve{\mathbf{B}})^{-1} = \mathcal{O}(1)$ as $p \to \infty$.

We also impose strong mixing condition. Let $\mathcal{F}_{-\infty}^{0}$ and $\mathcal{F}_{T}^{\infty}$ denote the σ-algebras that are generated by $\{(\mathbf{f}_t, \boldsymbol{\varepsilon}_t) : t \leq 0\}$ and $\{(\mathbf{f}_t, \boldsymbol{\varepsilon}_t) : t \geq T\}$ respectively. Define the mixing coefficient $\alpha(T) = \sup_{A \in \mathcal{F}_{-\infty}^{0}, B \in \mathcal{F}_{T}^{\infty}} |\Pr A \Pr B - \Pr AB|$.

(A.3) (Strong mixing) There exists $r_3 > 0$ such that $3r_1^{-1} + 1.5r_2^{-1} + 3r_3^{-1} > 1$, and $C > 0$ satisfying, for all $T \in \mathbb{Z}^+$, $\alpha(T) \leq \exp(-CT^{r_3})$.

Let $\mathbf{\Lambda}_q = \text{diag}(\lambda_1, \ldots, \lambda_q)$ be a matrix of q leading eigenvalues of $\mathbf{\Sigma}$, and $\mathbf{V}_q = (\mathbf{v}_1, \ldots, \mathbf{v}_q)$ is a $p \times q$ matrix of their corresponding leading eigenvectors. Define $\widehat{\mathbf{\Sigma}}, \widehat{\mathbf{\Lambda}}_q, \widehat{\mathbf{V}}_q$ to be the estimators of $\mathbf{\Sigma}, \mathbf{\Lambda}_q, \mathbf{V}_q$. We further let $\widehat{\mathbf{\Lambda}}_q = \text{diag}(\hat{\lambda}_1, \ldots, \hat{\lambda}_q)$ and $\widehat{\mathbf{V}}_q = (\hat{\mathbf{v}}_1, \ldots, \hat{\mathbf{v}}_q)$ to be constructed by the first q leading empirical eigenvalues and the corresponding eigenvectors

of $\widehat{\mathbf{\Sigma}}$ and $\widehat{\mathbf{B}}\widehat{\mathbf{B}}' = \widehat{\mathbf{V}}_q\widehat{\mathbf{\Lambda}}_q\widehat{\mathbf{V}}_q'$. Similarly to Fan et al. (2018), we require the following bounds on the componentwise maximums of the estimators:

(B.1) $\left\|\widehat{\mathbf{\Sigma}} - \mathbf{\Sigma}\right\|_{\max} = \mathcal{O}_P(\sqrt{\log p/T})$,

(B.2) $\left\|(\widehat{\mathbf{\Lambda}}_q - \mathbf{\Lambda}_q)\mathbf{\Lambda}_q^{-1}\right\|_{\max} = \mathcal{O}_P(\sqrt{\log p/T})$,

(B.3) $\left\|\widehat{\mathbf{V}}_q - \mathbf{V}_q\right\|_{\max} = \mathcal{O}_P(\sqrt{\log p/(Tp)})$.

Assumptions **(B.1)**-**(B.3)** are needed in order to ensure that the first q principal components are approximately the same as the columns of the factor loadings. The estimator $\widehat{\mathbf{\Sigma}}$ can be thought of as any "pilot" estimator that satisfies **(B.1)**. For sub-Gaussian distributions, sample covariance matrix, its eigenvectors and eigenvalues satisfy **(B.1)**-**(B.3)**.

In addition, the following structural assumption on the model is imposed:

(C.1) $\|\mathbf{\Sigma}\|_{\max} = \mathcal{O}(1)$ and $\|\mathbf{B}\|_{\max} = \mathcal{O}(1)$.

To study the properties of the combination weights and MSFE, we use the convergence properties of precision matrix produced by Algorithm B.2 established in Lee and Seregina (2024). Let $\omega_T \equiv \sqrt{\log p/T} + 1/\sqrt{p}$. Also, let $s(\mathbf{\Theta}_\varepsilon) = \mathcal{O}_P(s_T)$ for some sequence $s_T \in (0, \infty)$ and $d(\mathbf{\Theta}_\varepsilon) = \mathcal{O}_P(d_T)$ for some sequence $d_T \in (0, \infty)$. The deterministic sequences s_T and d_T will control the sparsity $\mathbf{\Theta}_\varepsilon$ for FGL. Note that d_T can be smaller than or equal to s_T.

Let ϱ_T be a sequence of positive-valued random variables such that $\varrho_T^{-1}\omega_T \xrightarrow{P} 0$ and $\varrho_T d_T s_T \xrightarrow{P} 0$, with $\tau \asymp \omega_T$ (where τ is the tuning parameter for the FGL in (B.1)). Lee and Seregina (2024) show that under the Assumptions **(A.1)**-**(A.3)**, **(B.1)**-**(B.3)** and **(C.1)**, $\left\|\left\|\widehat{\mathbf{\Theta}} - \mathbf{\Theta}\right\|\right\|_1 = \mathcal{O}_P(\varrho_T d_T s_T) = o_P(1)$ and $\left\|\left\|\widehat{\mathbf{\Theta}} - \mathbf{\Theta}\right\|\right\|_2 = \mathcal{O}_P(\varrho_T s_T) = o_P(1)$ for FGL.

Theorem 1 summarizes consistency results for the combination weights and the resulted MSFE.

Theorem 1. *Assume* ***(A.1)-(A.3)****,* ***(B.1)-(B.3)****, and* ***(C.1)*** *hold. FGL consistently estimates forecast combination weights and MSFE*$(\widehat{\mathbf{w}}, \widehat{\mathbf{\Sigma}})$*:*

(i) If $\varrho_T d_T^2 s_T \xrightarrow{P} 0$, $\|\widehat{\mathbf{w}} - \mathbf{w}\|_1 = \mathcal{O}_P\Big(\varrho_T d_T^2 s_T\Big) = o_P(1)$.

(ii) If $\varrho_T d_T s_T \xrightarrow{P} 0$, $\left|\frac{MSFE(\widehat{\mathbf{w}},\widehat{\mathbf{\Sigma}})}{MSFE(\mathbf{w},\mathbf{\Sigma})} - 1\right| = \mathcal{O}_P(\varrho_T d_T s_T) = o_P(1)$.

The proof of Theorem 1 can be found in Appendix C.[3] Note that the rate of convergence for MSFE is faster than the combination weight rate. In contrast to the classical graphical model in Algorithm A.1, the convergence properties of which were examined by Janková and van de Geer (2018) among others, the rates in Theorem 1 depend on the sparsity of $\mathbf{\Theta}_\varepsilon$ rather than of $\mathbf{\Theta}$. This means that instead of assuming that many partial correlations of forecast errors $\mathbf{e}_t$ are negligible, which is not realistic under the factor structure, we impose a milder restriction requiring many partial correlations of $\boldsymbol{\varepsilon}_t$ to be negligible once the common components have been taken into account.

3 RD-FGL for Forecast Errors

We augment the framework in Section 2 to account for regime switching by modelling the change in precision matrix due to N structural breaks. Macroeconomic and financial datasets typically span a long time period, hence, the assumptions of time-invariant factor loadings and constant idiosyncratic precision matrix are restrictive (Corradi and Swanson (2014)). Define $n_j \equiv t_j - t_{j-1}$ to be the sample between the j-th and $(j-1)$-th break points, where $j = 1, \ldots, N+1$, $\sum_{j=1}^{N+1} n_j = T$, $t_0 = 0$, $N \leq T$. Also, let $\mathbf{e}_t(\lambda) = \lambda^{T-t}\mathbf{e}_t$ be down-weighted observations, where $0 < \lambda \leq 1$ is the down-weighting coefficient. As suggested in

[3] As shown in extensive simulations conducted by Lee and Seregina (2024), FGL is robust to elliptical distributions. The assumptions **(A.1)-(A.3)**, **(B.1)-(B.3)**, and **(C.1)** are general enough to accommodate distributions with heavier tails.

Chudik et al. (2024) and Pesaran et al. (2013), down-weighting is beneficial for forecasting since it additionally weighs recent observations more than very distant ones.

3.1 Regime-Dependent Factor Loadings

As a first extension to FGL, we model structural changes in factor loadings using a framework similar to Su and Wang (2017). For now assume a single known break $N = 1$ which occurs at T_1. Write equation (2.1) as:

$$e_{it}(\lambda) = \underbrace{\mathbf{b}_i'}_{1\times q}\underbrace{\mathbf{f}_t}_{q\times 1} + \varepsilon_{it}, \quad t = 1, \dots, T, \ i = 1, \dots, p. \tag{3.1}$$

To estimate $\{\mathbf{b}_i\}_{i=1}^p$ and $\{\mathbf{f}_t\}_{t=1}^T$, we can consider the following weighted least squares problem:

$$\min_{\{\mathbf{b}_i\}_{i=1}^p, \{\mathbf{f}_t\}_{t=1}^T} (pT)^{-1} \sum_{i=1}^{p}\sum_{t=1}^{T} \left[e_{it}(\lambda) - \mathbf{b}_i'\mathbf{f}_t\right]^2 K_{\gamma t}, \tag{3.2}$$

subject to certain identification restrictions to be specified later on. Here, $K_{\gamma t} = \gamma \mathbb{1}\left[t \leq T_1\right] + \mathbb{1}\left[t > T_1\right]$ is a discrete kernel as in Li et al. (2013) with $\gamma \in [0, 1]$. Since more recent information is usually more relevant to forecasting, such kernel-weight estimator gives weight 1 to post-break observations and weight γ to pre-break observations.

Define the $T \times p$ matrices $\mathbf{E}(\lambda, \gamma) = \Big(\mathbf{e}_1(\lambda, \gamma), \dots, \mathbf{e}_p(\lambda, \gamma)\Big)$, $\mathcal{E}(\lambda, \gamma) = \Big(\boldsymbol{\varepsilon}_1(\lambda, \gamma), \dots, \boldsymbol{\varepsilon}_p(\lambda, \gamma)\Big)$, where $\mathbf{e}_i(\lambda, \gamma) = \Big(K_{\gamma 1}^{1/2} e_{i1}(\lambda), \dots, K_{\gamma T}^{1/2} e_{iT}(\lambda)\Big)'$ and $\boldsymbol{\varepsilon}_i(\lambda, \gamma) = \Big(K_{\gamma 1}^{1/2} \varepsilon_{i1}(\lambda), \dots, K_{\gamma T}^{1/2} \varepsilon_{iT}(\lambda)\Big)'$. Also, let $\mathbf{F}(\lambda, \gamma) = \Big(K_{\gamma 1}^{1/2} \mathbf{f}_1(\lambda), \dots, K_{\gamma T}^{1/2} \mathbf{f}_T(\lambda)\Big)'$ be a $T \times q$ matrix collecting factors. In matrix notation, the transformed model in (3.1) can be written as $\mathbf{E}(\lambda, \gamma) = \mathbf{F}(\lambda, \gamma)\mathbf{B}' + \mathcal{E}(\lambda, \gamma)$, where $\mathbf{B} = (\mathbf{b}_1, \dots, \mathbf{b}_p)'$ is a $p \times q$ matrix of factor loadings.

As shown in Su and Wang (2017) for the continuous kernel, the minimization problem

in (3.2) reduces to:

$$\min_{\mathbf{F}(\lambda,\gamma),\mathbf{B}} \text{tr}\left[\Big(\mathbf{E}(\lambda,\gamma)-\mathbf{F}(\lambda,\gamma)\mathbf{B}'\Big)\Big(\mathbf{E}(\lambda,\gamma)-\mathbf{F}(\lambda,\gamma)\mathbf{B}'\Big)'\right] \tag{3.3}$$
$$\text{s.t. } \mathbf{F}'(\lambda,\gamma)\mathbf{F}(\lambda,\gamma)/T=\mathbf{I}_q \text{ and } \mathbf{B}'\mathbf{B}=\text{diagonal matrix.}$$

The problem in (3.3) is the conventional PCA problem. The estimated factor matrix $\widehat{\mathbf{F}}(\lambda,\gamma) = \Big(K_{\gamma 1}^{1/2}\widehat{\mathbf{f}}_1(\lambda),\ldots,K_{\gamma T}^{1/2}\widehat{\mathbf{f}}_T(\lambda)\Big)'$ is $\sqrt{T}$ times eigenvectors corresponding to the q largest eigenvalues of $\mathbf{E}(\lambda,\gamma)\mathbf{E}'(\lambda,\gamma)$, arranged in descending order, and $\widehat{\mathbf{B}}'(\lambda,\gamma) = (\widehat{\mathbf{F}}(\lambda,\gamma)\widehat{\mathbf{F}}'(\lambda,\gamma))^{-1}\widehat{\mathbf{F}}'(\lambda,\gamma)\mathbf{E}(\lambda,\gamma) = \widehat{\mathbf{F}}'(\lambda,\gamma)\mathbf{E}(\lambda,\gamma)/T$ are the estimators of the corresponding time-varying factor loadings, where $\widehat{\mathbf{B}}(\lambda,\gamma) = (\widehat{\mathbf{b}}_1(\lambda,\gamma),\ldots,\widehat{\mathbf{b}}_p(\lambda,\gamma))'$ is $p\times q$.

Since the estimator $\widehat{\mathbf{F}}(\lambda,\gamma)$ is only consistent up to a rotation, we use a two-stage estimation procedure to obtain a consistent estimator (Su and Wang (2017)). Based on the consistent estimators of $\mathbf{b}_i$'s obtained from the first stage, consistent estimators of $\mathbf{f}_t(\lambda,\gamma)$ can be obtained by considering the following least squares problem $\widehat{\mathbf{f}}_t(\lambda,\gamma) = \text{argmin}_{\mathbf{f}_t}\sum_{i=1}^p\Big[e_{it}(\lambda)-\widehat{\mathbf{b}}_i'(\lambda,\gamma)\mathbf{f}_t\Big]^2$ which yields the solution $\widehat{\mathbf{f}}_t(\lambda,\gamma) = \Big(\sum_{i=1}^p\widehat{\mathbf{b}}_i(\lambda,\gamma)\widehat{\mathbf{b}}_i'(\lambda,\gamma)\Big)^{-1}\Big(\sum_{i=1}^p\widehat{\mathbf{b}}_i(\lambda,\gamma)e_{it}(\lambda)\Big)$.

Remark 1. As in Su and Wang (2017), we assume that $\mathbb{E}[\mathbf{f}_t\mathbf{f}_t']$ is homogeneous over t. This assumption is not restrictive, since if $\mathbb{E}[\mathbf{f}_t\mathbf{f}_t'] = \mathbf{\Sigma}_{f,t}$, we can rewrite the common component as $\mathbf{b}_i'\mathbf{f}_t = \Big(\mathbf{\Sigma}_f^{-1/2}\mathbf{\Sigma}_{f,t}^{1/2}\mathbf{b}_i\Big)'\mathbf{\Sigma}_f^{1/2}\mathbf{\Sigma}_{f,t}^{-1/2}\mathbf{f}_t = \mathbf{b}_i^{*\prime}\mathbf{f}_t^{*\prime}$, where $\mathbf{b}_i^* = \mathbf{\Sigma}_f^{-1/2}\mathbf{\Sigma}_{f,t}^{1/2}\mathbf{b}_i$, and $\mathbf{f}_t^* = \mathbf{\Sigma}_f^{1/2}\mathbf{\Sigma}_{f,t}^{-1/2}\mathbf{f}_t$ satisfies $\mathbb{E}\left[\mathbf{f}_t^*\mathbf{f}_t^{*\prime}\right] = \mathbf{\Sigma}_f$ for each t.

To choose the optimal tuning parameters λ and γ in (3.2), we use the cross-validation

and solve the following minimization problem:

$$\min_{\gamma,\lambda} \mathrm{CV}(\gamma,\lambda) = \frac{1}{p(T-T_1)} \sum_{i=1}^{p} \sum_{s=T_1+1}^{T} \left[e_{is}(\lambda) - \widehat{\mathbf{b}}_i^{\prime(-s)}(\lambda,\gamma) \widehat{\mathbf{f}}_s^{(-s)}(\lambda,\gamma) \right]^2, \tag{3.4}$$

where $\widehat{\mathbf{b}}_i^{(-s)}(\lambda,\gamma)$ and $\widehat{\mathbf{f}}_s^{(-s)}(\lambda,\gamma)$ are estimated by leaving the s-th time series observation out of the PCA procedure.

Remark 2. The procedure for estimating regime-dependent factor loadings can be easily extended to the case when the number of breaks is greater than 1 ($N > 1$). The kernel in (3.2) would be adjusted accordingly $K_{\gamma_j t} = \gamma_j \mathbb{1}\left[t \leq T_j\right] + \mathbb{1}\left[t > T_N\right]$, where $j = 1, \ldots, N+1$. To estimate $\{\gamma_j\}_{j=1}^{N+1}$ we use cross-validation as in (3.4) consequently applied to each two periods separated by a break.

3.2 Regime-Dependent Idiosyncratic Precision Matrix

As a second extension to FGL, we model structural changes in the precision matrix of the idiosyncratic component. Let $\mathbf{\Sigma}_{\varepsilon,j}$ and $\mathbf{\Sigma}_j$ be covariance matrices of idiosyncratic part and forecast errors in regime j. Define the corresponding precision matrices to be $\mathbf{\Theta}_{\varepsilon,j} \equiv \mathbf{\Sigma}_{\varepsilon,j}^{-1}$ and $\mathbf{\Theta}_j \equiv \mathbf{\Sigma}_j^{-1}$. Similarly to the previous subsection, we assume $\mathbf{\Sigma}_{f_j} = \mathbf{\Sigma}_f$ for all regimes j.

Let $\widehat{\mathbf{\Sigma}}_{\varepsilon,j} = \frac{1}{n_j} \sum_{k=1}^{n_j} \widehat{\boldsymbol{\varepsilon}}_{j,k}(\lambda) \widehat{\boldsymbol{\varepsilon}}_{j,k}(\lambda)'$. To model dynamics in $\{\mathbf{\Theta}_{\varepsilon,j}\}_{j=1}^{N+1}$ we use the following optimization problem:

$$\min_{\{\mathbf{\Theta}_{\varepsilon,j}\}_{j=1}^{N+1}} \sum_{j=1}^{N+1} n_j \left[\mathrm{tr}\left(\widehat{\mathbf{\Sigma}}_{\varepsilon,j} \mathbf{\Theta}_{\varepsilon,j} \right) - \log\det \mathbf{\Theta}_{\varepsilon,j} \right] + \alpha \|\mathbf{\Theta}_{\varepsilon,j}\|_{\mathrm{od},1} + \beta \sum_{j=2}^{N+1} \psi(\mathbf{\Theta}_{\varepsilon,j} - \mathbf{\Theta}_{\varepsilon,j-1}), \tag{3.5}$$

where the penalty for the off-diagonal (od) elements is $\|\mathbf{\Theta}_{\varepsilon,j}\|_{\mathrm{od},1} = \sum_{l \neq q} \widehat{\gamma}_{\varepsilon,ll,j} \widehat{\gamma}_{\varepsilon,qq,j} |\theta_{\varepsilon,lq,j}|$, $\widehat{\gamma}_{\varepsilon,ll,j}$ is the (l,l)-th element of $\widehat{\mathbf{\Gamma}}_{\varepsilon,j}^2 \equiv \mathrm{diag}(\widehat{\mathbf{\Sigma}}_{\varepsilon,j})$ and $\theta_{\varepsilon,lq,j}$ is the lq-th element of matrix $\mathbf{\Theta}_{\varepsilon,j}$.

Figure 3 visualizes dynamics of the precision matrix.

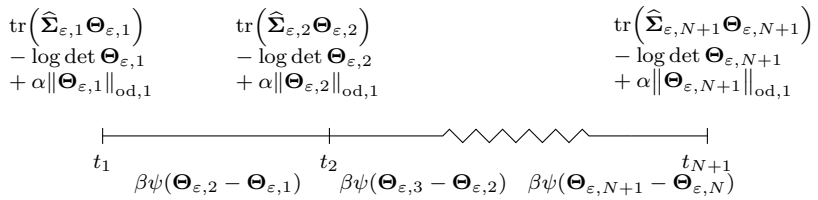

Figure 3: Change of precision matrix over time: β is the penalty that enforces temporal consistency and ψ is a convex penalty function.

The optimization problem in (3.5) has two tuning parameters: α, which determines the sparsity level of the network, and β, which controls the strength of resemblance between two neighboring precision estimators. In simulations and the empirical application we use the following procedure for tuning α and β: first, we set a grid of values $(\alpha, \beta) \in \{0, 0.25, 0.5, 1, 10, 30\}$. Second, we use the first 2/3 of the training data to estimate forecast combination weights and jointly tune α and β in the remaining 1/3 to yield the smallest value of the objective function, which is chosen to be either $\|\|\cdot\|\|_2$-loss of precision matrix or MSFE for simulations in Supplemental Appendix E and the empirical application. Note that when $\beta = 0$, the optimization in (3.5) reduces to estimating $\mathbf{\Theta}_{\varepsilon,i}$ using Algorithm B.2 in each regime separately. Naturally, this incorporates the case when the structural break is strong and only the post-break data is used for producing forecast combination weights. When β is large, there are weak structural breaks in $\mathbf{\Theta}_{\varepsilon,j}$, and $\mathbf{\Theta}_{\varepsilon,j}$'s are estimated by using the data across different regimes. Sections 4 and 5 provide more discussion on this in the context of our empirical application.

The smoothing function $\psi(\cdot)$ in (3.5) can be LASSO ($\psi(\cdot) = \sum_{l,q}|\cdot|$), Group LASSO ($\psi(\cdot) = \sum_q\|\cdot_q\|_2$), or Ridge ($\psi(\cdot) = \sum_{l,q}(\cdot_{lq})^2$). LASSO penalty encourages small changes in the precision matrix over time: when the lq-th element changes at two consecutive times, the

penalty forces the rest of the elements of the precision to remain the same. Group LASSO penalty allows the entire graph to restructure at some time points. This penalty is useful for anomaly detection, since it can identify structural changes in the network structure. Ridge penalty allows the network to change smoothly over time. This penalty is less strict than the LASSO penalty: instead of encouraging the graphs to be exactly the same, it allows smooth transitions.

To estimate (3.5) we use the ADMM algorithm described in details in the Online Supplement S1. Once $\boldsymbol{\Theta}_{\varepsilon,i}$ is estimated, we combine estimated factors, loadings and precision matrix of the idiosyncratic components using the Sherman-Morrison-Woodbury formula to estimate the final precision matrix of forecast errors and use it to compute optimal forecast combination weights. We call the aforementioned procedure RD-FGL and summarize it in Algorithm 1.

Algorithm 1 RD-FGL

1: Estimate $\{\mathbf{b}_i\}_{i=1}^{p}$ and $\{\mathbf{f}_t(\lambda,\gamma_j)\}_{t=1}^{T}$ in (3.1) using the weighted least squares problem in (3.2). Get $\widehat{\boldsymbol{\Sigma}}_f$, $\widehat{\boldsymbol{\Theta}}_f$ and $\widehat{\boldsymbol{\varepsilon}}_t(\lambda,\gamma_j) = \mathbf{e}_t(\lambda) - \widehat{\mathbf{B}}(\lambda,\gamma_j)\widehat{\mathbf{f}}_t(\lambda,\gamma_j)$.

2: Solve (3.5) using ADMM to get $\widehat{\boldsymbol{\Theta}}_{\varepsilon,j}$.

3: Use $\widehat{\boldsymbol{\Theta}}_{\varepsilon,j}$, $\widehat{\boldsymbol{\Theta}}_f$ and $\widehat{\mathbf{B}}(\lambda,\gamma_j)$ from Steps 1-2 to get $\widehat{\boldsymbol{\Theta}}_j(\lambda,\gamma_j) = \widehat{\boldsymbol{\Theta}}_{\varepsilon,j} - \widehat{\boldsymbol{\Theta}}_{\varepsilon,j}\widehat{\mathbf{B}}(\lambda,\gamma_j)[\widehat{\boldsymbol{\Theta}}_f + \widehat{\mathbf{B}}'(\lambda,\gamma_j)\widehat{\boldsymbol{\Theta}}_{\varepsilon,j}\widehat{\mathbf{B}}(\lambda,\gamma_j)]^{-1}\widehat{\mathbf{B}}'(\lambda,\gamma_j)\widehat{\boldsymbol{\Theta}}_{\varepsilon,j}$.

4: Use $\widehat{\boldsymbol{\Theta}}_j(\lambda,\gamma_j)$ to get forecast combination weights $\widehat{\mathbf{w}}_j(\lambda,\gamma_j) = \frac{\widehat{\boldsymbol{\Theta}}_j(\lambda,\gamma_j)\boldsymbol{\iota}_p}{\boldsymbol{\iota}_p'\widehat{\boldsymbol{\Theta}}_j(\lambda,\gamma_j)\boldsymbol{\iota}_p}$.

We develop a scalable implementation of (3.5) for the RD-FGL in Algorithm 1 through ADMM, which is extensively discussed in the Online Supplement S1. ADMM is a distributed convex optimization approach (Parikh and Boyd (2014)) that allows us to split the optimization problem in (3.5) into a series of subproblems. As pointed out in Hallac et al.

(2017), the scalability of ADMM comes from the improved runtime: to estimate a $p \times p$ matrix, the cost per iteration of ADMM is $\mathcal{O}(p^3)$ (which is the cost of an eigendecomposition of the $\mathbf{\Theta}$ step in the Online Supplement S1). In contrast, the runtime of general interior-point methods is $\mathcal{O}(p^6)$ (Mohan et al. (2014)).

Remark 3. Let us comment on the theoretical properties of RD-FGL. First, as shown in Su and Wang (2017), introducing time-varying factors does not change the main assumptions **(A.1)**-**(A.3)** on the errors, factors, factor loadings, and their interactions. This is because, as shown in (3.3), the formulation with time-varying loadings can be reduced to the conventional PCA problem. Additional assumption that we need to impose is that $\mathbb{E}[\mathbf{f}_t\mathbf{f}_t']$ is homogeneous over t. As discussed in Subsection 4.1, this assumption is not restrictive. Second, we assume that the number of factors, q, and the number of forecasts, p, are not affected by the structural changes in loadings or idiosyncratic precision matrix. Allowing p and q to change is a straightforward extension and is left for future research. Third, assumptions **(B.1)**-**(B.3)** and assumption **(C.1)** are required to hold for each regime $j = 1, \ldots, N+1$. Finally, we allow $s(\mathbf{\Theta}_{\varepsilon,j}) = \mathcal{O}_P(s_{n_j})$ and $d(\mathbf{\Theta}_{\varepsilon,j}) = \mathcal{O}_P(d_{n_j})$ to change for $j = 1, \ldots, N+1$. Let $\omega_{n_j} \equiv \sqrt{\log p / n_j} + 1/\sqrt{p}$. As long as $\varrho_{n_j}^{-1}\omega_{n_j} \xrightarrow{P} 0$ and $\varrho_{n_j} d_{n_j} s_{n_j} \xrightarrow{P} 0$ for each j, RD-FGL achieves the same rate as FGL in each regime:

(i) If $\varrho_{n_j} d_{n_j}^2 s_{n_j} \xrightarrow{\mathrm{P}} 0$, RD-FGL consistently estimates forecast combination weights $\widehat{\mathbf{w}}_j(\lambda, \gamma_j)$ in Algorithm 1: $\|\widehat{\mathbf{w}}_j(\lambda, \gamma_j) - \mathbf{w}_j\|_1 = \mathcal{O}_P\left(\varrho_{n_j} d_{n_j}^2 s_{n_j}\right) = o_P(1)$.

(ii) If $\varrho_{n_j} d_{n_j} s_{n_j} \xrightarrow{\mathrm{P}} 0$, FGL consistently estimates $\mathrm{MSFE}(\mathbf{w}_j, \mathbf{\Sigma}_j)$: $\left|\frac{\mathrm{MSFE}(\widehat{\mathbf{w}}_j(\lambda,\gamma_j), \widehat{\mathbf{\Sigma}}_j)}{\mathrm{MSFE}(\mathbf{w}_j, \mathbf{\Sigma}_j)} - 1\right| = \mathcal{O}_P(\varrho_{n_j} d_{n_j} s_{n_j}) = o_P(1)$.

3.3 Unknown Break Time and Number of Breaks

The previous two subsections assumed that the number and location of breaks are known. We now relax these assumptions. First, assume that the number of breaks in factor loadings, N_B, and the number of breaks in idiosyncratic precision, $N_{\mathbf{\Theta}}$, are known and $N_B = N_{\mathbf{\Theta}} = 1$, but their locations are unknown and might differ from each other.

To estimate the location of the break in factor loadings, we adapt the procedure in Bai et al. (2020). For a given break point in loadings, T_1, define the sum of squared residuals (SSR) as in (3.2):

$$\text{SSR}(T_1) = (pT)^{-1} \sum_{i=1}^{p} \sum_{t=1}^{T} \left[e_{it}(\lambda) - \mathbf{b}_{it}' \mathbf{f}_t \right]^2 K_{\gamma t}, \tag{3.6}$$

where $K_{\gamma t} = \gamma \mathbb{1}\left[t \leq T_1\right] + \mathbb{1}\left[t > T_1\right]$ is a discrete kernel. The estimated break date is given by $\widehat{T}_1 = \text{argmin}_{1 \leq T_1 \leq T-1} \text{SSR}(T_1)$.

To estimate the location of the break in $\mathbf{\Theta}_\varepsilon$ we use the procedure similar to Bai (2010). Define t_1 to be a break point in $\mathbf{\Theta}_\varepsilon$. Recall, $n_j = t_j - t_{j-1}$, where $j = 1, 2$. Note that the number of observations in each regime depends on t_1: $n_1 = t_1 - t_0$ and $n_2 = t_2 - t_1$. For a given break point in idiosyncratic precision t_1, define the following objective function as in (3.5):

$$L(t_1) = \sum_{j=1}^{2} n_j \left[\text{tr}\left(\widehat{\mathbf{\Sigma}}_{\varepsilon,j} \mathbf{\Theta}_{\varepsilon,j} \right) - \log\det \mathbf{\Theta}_{\varepsilon,j} \right] + \alpha \| \mathbf{\Theta}_{\varepsilon,j} \|_{\text{od},1} + \beta \psi(\mathbf{\Theta}_{\varepsilon,2} - \mathbf{\Theta}_{\varepsilon,1}). \tag{3.7}$$

The estimated break date is given by $\widehat{t}_1 = \text{argmin}_{1 \leq t_1 \leq T-1} L(t_1)$.

When the number of breaks in either loadings is known and greater than 1 ($N_B > 1$) and/or $N_{\mathbf{\Theta}} > 1$, we can use the one-at-a-time approach as in Bai (2010): the objective functions are identical to (3.6) and (3.7). The breaks are estimated sequentially. Once the

first break is obtained, we split the sample at the estimated break point, resulting in two subsamples. A single break point in each subsample is estimated, but only one that achieves the smallest objective function ((3.6) or (3.7)) is retained. If the number of breaks is equal to two, the procedure is stopped. Otherwise, we continue splitting into subsamples until all breaks are estimated.

If the number of breaks is unknown, we proceed as suggested in Bai (2010): in the aforementioned one-at-a-time approach apply the test for existence of break point (Bai and Perron (2003)) to each subsample before estimating a break point.

We refer an interested reader to Supplemental Appendix E that provides extensive simulation results examining the performance of RD-FGL. To summarize the findings: we confirm that RD-FGL consistently estimates precision matrix, forecast combination weights, and MSFE. The results are robust to different strengths of common factors, presence of multiple breaks, and varying break magnitude.

Remark 4. (a) Defining regimes by a time break assumed that a regime cannot be revisited again. However, in the context of co-movements across forecasters, the co-movement might be especially strong in certain periods (e.g., around a crisis). To incorporate a possibility of revisiting a regime, we define S to be a set that determines a regime. Define $K_{\gamma,t} = \gamma \mathbb{1}[t \in S] + \mathbb{1}[t \notin S]$ with $\gamma \in \mathbb{R}$. The weighted least squares problem in (3.2) remains unchanged, only the kernel function is modified.

(b) For precision matrix, let $N_{\boldsymbol{\Theta}} = 1$, which corresponds to two states, n_j are the observations that belong to $t \in S$ (which will be denoted as Regime $j = 1$), and $t \notin S$ (Regime $j = 2$), $\sum_{j=1}^{2} n_j = T$, $t_0 = 0$. Define $\widehat{\boldsymbol{\Sigma}}_{\varepsilon,j} = \frac{1}{n_j} \sum_{k=1}^{n_j} \widehat{\boldsymbol{\varepsilon}}_{j,k}(\lambda) \widehat{\boldsymbol{\varepsilon}}_{j,k}(\lambda)'$. Then (3.5) remains unchanged, only the way how regimes are defined changes. Similarly to subsection 3.1 (as noted in Remark 2), the aforementioned setup can be easily extended to include more than 2 regimes.

(c) We consider the states determined by a simple self-exciting threshold autoregressive model (SETAR). Specifically, in Sections 4 and 5, we let $S \equiv \{t : y_t > \bar{y}_t\}_{t=1}^{T}$ be a set that defines a regime, where y_t is the target series and $\bar{y}_t$ is the average of the target series over τ previous periods. We choose τ to match the frequency of the dataset. One could also estimate τ using threshold approaches. If we were to determine the states in general, it would involve a choice of (exogenous) thresholding variables which is beyond the scope of the paper.

4 Application to Combining ECB SPF Forecasts

We use quarterly forecasts on the expected rates of inflation, real GDP growth and unemployment rate in the Euro area published by the ECB. The raw data records 119 forecasters in total, but the panel is highly unbalanced with many missing values due to entry and exit in the long span. To obtain most qualified forecasters we proceed as follows: first, we filter out irregular respondents if they missed more than 45% of the observations; second, we use a random forest imputation algorithm (Stekhoven (2022); Stekhoven and Buhlmann (2012)) to interpolate the remaining missing values. We consider the forecasts of three main economic indicators: (1) Real GDP growth defined as the year-on-year (YoY) percentage change of real GDP, based on standardized European System of National and Regional Accounts definition. The time period under consideration is 1999:Q3-2024Q1 (which yields the total number of observations equal to 99), the final number of forecasters is $p = 57$, and the prediction horizon is 2-quarters ahead. (2) Inflation which is defined as the YoY percentage change of the Harmonised Index of Consumer Prices (HICP) published by Eurostat. The time period under consideration is 2000:Q1-2024Q2 (which yields the total number of observations equal to 98), the final number of forecasters is $p = 59$, and the prediction horizon

is 2-quarters ahead. (3) Unemployment rate which refers to Eurostat's definition and it is calculated as percentage of the labor force. The time period under consideration is 2000:Q1-2024Q2 (which yields the total number of observations equal to 98), the final number of forecasters is $p = 46$, and the prediction horizon is 2-quarters ahead.

We consider three choices of the training sample: $R \in \{30, 40, 50\}$, the estimation window is rolled over the test sample to update the estimates at each point of time. We first estimate time-varying mean of the forecast errors $\mathbf{m}_t$ using AR(1). We use the estimated $\widehat{\mathbf{m}}_t$ to demean forecast errors. After that we estimate a factor model. The optimal number of factors in the forecast errors (denoted as q in equation (2.1)) is chosen using the standard data-driven method that uses the information criterion IC1 described in Bai and Ng (2002). In the majority of the cases the optimal number of factors was estimated to be equal to 1 or 2. To explore the benefits of using FGL and RD-FGL for forecast error quantification, we consider several alternative estimators of covariance/precision matrix of the idiosyncratic component in equation (2.5): (1) linear shrinkage estimator of covariance developed by Ledoit and Wolf (2004) further referred to as Factor LW (FLW); (2) nonlinear shrinkage estimator of covariance by Ledoit and Wolf (2017) (Factor NLW or FNLW); (3) POET (Fan et al. (2013)); (4) constrained ℓ_1-minimization for inverse matrix estimator, CLIME (Cai et al. (2011)) (Factor CLIME or FCLIME); (5) nodewise regression developed by Meinshausen and Bühlmann (2006) (Factor MB or FMB). To examine the benefits of imposing sparsity on $\mathbf{\Theta}_\varepsilon$ we also include the factor model without sparsity assumption on the idiosyncratic error precision matrix (referred to as Not Sparse) – this corresponds to imposing $\tau = 0$ in (B.1). To examine the benefits of using factor structure[4], we include several counterparts of the aforementioned models that directly estimate precision of the

[4]As an alternative to PCA, Boot and Nibbering (2019) use random subspace methods.

forecast errors without estimating factors and loadings: GL, LW, NLW, CLIME, and MB. We also include a univariate AR(1) model of the target series, which is the model that is not based on the combinations of forecasters. For RD-FGL, since different specifications of the smoothing function $\psi(\cdot)$ were shown to perform similarly in the simulations, we only keep ℓ_2-penalty. The break parameter in factor loadings is estimated using cross-valiation ($\gamma = \hat{\gamma}$), whereas the down weighting parameter λ is fixed at 0.98 as recommended by Chudik et al. (2024). We consider two types of a breaks: breaks determined by a time point, and breaks determined by a state that defines a regime. For time breaks, the location and the number of breaks are estimated via the procedures discussed in subsection 3.3. For state switches we proceed as follows: let $S \equiv \{t : y_t > \bar{y}_t\}_{t=1}^{T}$ be a set that defines a regime, where y_t is the target series. Since ECB SPF has quarterly frequency, we use $\bar{y}_t = \frac{1}{4}\sum_{t-5}^{t-1} y_t$, which is the average of the target series over the previous four quarters. We consider two regimes: $t \in S$ and $t \notin S$. We note that the aforementioned definition of S does not attempt to separate between the periods of recession and expansion. One can certainly come up with a more elaborate definition of regimes for such purposes. Our goal is to see if the possibility of revisiting a regime, which was not present for time breaks, can be useful when combining forecasts. To check whether the proposed regimes align with periods of higher forecast uncertainty, we computed correlations between average values of forecast errors and the state variable (defined as $y_t - \bar{y}_t$). These are reported in Figure F.6 of the Supplementary Appendix together with a graph of forecast errors and the state variable. The correlation values are 0.88, 0.93, and 0.59 for Real GDP, HICP, and Unemployment Rate, respectively. The magnitude of correlations suggests that the large values of the state variable align with periods of large forecast errors.

Our benchmark is the simple average with equal weights on all forecasters (referred to as

EW), since it is a commonly used benchmark (see Genre et al. (2013) and Thompson et al. (2024) among others). Going back to the discussion in Section 3 regarding setting $\beta = 0$ in equation (3.5): as we pointed out, this corresponds to using only post-break sample for estimation which is suboptimal since the value of β is already chosen optimally from the grid that includes $\beta = 0$ to minimize the MSFE. Hence, by construction, RD-FGL is superior to using only post-break data.

For RD-FGL with time breaks the number of breaks for loadings and precision is estimated using the test for existence of break point (Bai and Perron (2003)): using their sequential procedure we search for up to three breaks and set the trimming parameter to 10% of the total number of observations, and the significance level at 5%. The location of the break points for each series is estimated using the one-at-a-time approach described in Subsection 3.3.

Table 1 compares the performance of FGL and RD-FGL with the competitors for predicting three macroeconomic indicators for Euro-area using a combination of ECB SPF forecasts. It reports the ratios of MSFE of each method to the MSFE of the EW. Using the Model Confidence Set (MCS) of Hansen et al. (2011), we identify the set of superior models (SSM) for each series and horizon at 90% confidence level. Models marked with a star in Table 1 belong to the SSM according to MCS test which ranks them according to the relative sample loss of the i-th model relative to the average across models in SSM.

There are five main findings that we learn from analysing Table 1: **(1)** for most series factor-based models outperform non-factor ones. This means that incorporating the factor structure in the forecast errors improves forecasting performance. **(2)** for all series the model based on sample covariance provides one of the worst performances. This means that the factor structure per se is not sufficient to achieve performance gains over EW, hence, it is

Table 1: Prediction of Quarterly Macroeconomic Variables for Euro-area Using ECB SPF Forecasts.

	AR1	**GL**	**LW**	**NLW**	**POET**	**CLIME**	**MB**	**Not Sparse**	**FGL**	**FLW**	**FNLW**	**FCLIME**	**FMB**	**RD-FGL (time)**	**RD-FGL (state)**
Real GDP growth															
$R = 30$	3.1773	2.4667	1.1795	2.5876	0.9048*	2.2085	1.7829	19.3187	0.9618	1.1303	2.4148	0.9979	0.8865*	**0.8543***	0.9372*
$R = 40$	4.9930	1.1737	1.4431	4.3341	5.3961	2.1275	1.0216	53.8817	1.0644	1.3206	2.4331	0.9980	0.9756	**0.9431***	0.9545*
$R = 50$	1.8101	1.1781	1.0260	2.6964	1.0941	1.7791	1.1817	51.9385	1.0019	1.2157	1.8525	0.9986	1.0009	**0.9564***	0.9580*
Inflation															
$R = 30$	**0.7070***	0.9659	0.8660	1.4350	0.8785	0.9822	0.9227	1.1978	0.9874	0.8346	0.8750	0.9651	0.8754	0.8568	0.7290*
$R = 40$	1.3240	1.0150	1.0512	0.9300	0.8449*	1.0923	1.4599	1.3154	0.9680	0.9587	1.1163	1.4599	1.3323	0.9283	**0.7818***
$R = 50$	1.8276	1.1873	0.9895	1.4266	1.5702	1.2969	1.2310	2.7828	1.0999	**0.9223***	1.1857	1.2310	1.2958	1.2079	1.1612
Unemployment rate															
$R = 30$	1.1982	0.7776*	0.8084*	0.7890*	0.9101	0.8273*	14.5622	1.5102	**0.7640***	0.7956*	0.7824*	0.9161	1.1019	1.0381	0.8920
$R = 40$	1.4236	1.1161	1.4208	1.4869	1.6948	1.1340	2.7885	7.9007	0.9819*	1.1595	1.1528	**0.9772***	1.1847	1.4777	1.3051
$R = 50$	1.3631	0.9781	1.2776	1.5103	1.5593	1.1146	3.9806	10.9871	**0.8086***	1.1804	1.0807	0.9433	2.2911	1.7878	1.6067

Note: MSFEs of competing methods are reported for each value of R, where R indicates the length of the training window. RD-FGL (time) is our method with a time-break, and RD-FGL (state) is our method with a state switch. Ratio indicates the ratio to MSFE of the Equal-Weighted combined forecast. Models with the lowest ratio are in bold. Models marked with a star belong to the SSM according to MCS test which ranks them according to the relative sample loss of the i-th model relative to the average across models in SSM (at 90% confidence level).

necessary to impose sparsity on the precision matrix of the idiosyncratic components. **(3)** For real GDP growth and inflation series RD-FGL is included in the SSM for most values of R. For the unemployment rate, FGL outperforms RD-FGL. This result is supported by the behaviour, observed in the actual series: real GDP growth and inflation exhibit strong breaks following the global financial crisis and Covid pandemic, however this is not the case for the unemployment rate series that did not have strong breaks throughout the whole sample period. **(4)** For real GDP growth RD-FGL with time breaks and state switches perform comparatively similar. For inflation and unemployment series RD-FGL with state switches outperforms the one with time breaks. This highlights the benefits of allowing a possibility of recurring regimes. **(5)** The majority of combination models outperform AR(1) which emphasizes the benefits of combining multiple forecasts.

5 Application to FRED-MD

In a second empirical application, we use a large monthly frequency macroeconomic database of McCracken and Ng (2016), who provide a comprehensive description of the dataset and 126 macroeconomic series. We consider the time period 1959:1-2024:08 with the total number of observations $T = 788$.[5] We use four forecast targets: monthly industrial production (INDPRO), consumer price index for all items (CPIAUCSL), personal consumer expenditures: Chain Index (PCEPI), and civilian Unemployment Rate (UNRATE). To create forecasts we use the optimal combination of target series forecasts (see equation (2.3)) based on individual monthly indicators.[6] We split the sample in three parts: the first 240 observations are used to estimate the competing forecasting models. The remaining 548 ob-

[5] We use "current vintage" as of 2024:11.

[6] We thank the associate editor for this suggestion.

servations are split into the training sample $m = 400$, and the test sample $n = T - m - h + 1$, where h is the forecast horizon. We roll the estimation window over the test sample to update all the estimates in each point of time $t = m, \ldots, T - h$. For INDPROD, CPIAUCSL, and PCEPI we forecast the average growth rate (series Y_t is in logarithms), UNRATE we forecast the average change (series Y_t is without logarithms):

$$y_{t+h}^{(h)} = \frac{1}{h}(Y_{t+h}/Y_t).$$

The total number of forecasting models is $p = 125$. Similarly to Section 4, we first estimate time-varying mean $\mathbf{m}_t$ of each component of forecast errors $\tilde{e}_t$ using univariate AR(1) models and use it to demean forecast errors. After that we estimate factor model. The optimal number of factors in the forecast errors is chosen using the standard data-driven method that uses the information criterion IC1 described in Bai and Ng (2002).

We use the same alternative estimators of precision/covariance matrix as in Section 4. The break parameter in factor loadings is estimated using cross-valiation ($\gamma = \hat{\gamma}$), whereas the down weighting parameter λ is fixed at 0.98 as recommended by Chudik et al. (2024). We use RD-FGL with a state switch defined by $S \equiv \{t : y_t > \bar{y}_t\}_{t=1}^{T}$, where y_t is the target series. Since FRED-MD has monthly frequency, we use $\bar{y}_t = \frac{1}{12}\sum_{t-13}^{t-1} y_t$, which is the average of the target series over the previous twelve months. We consider two regimes: $t \in S$ and $t \notin S$. Similarly to the ECB SPF application, we check whether the proposed regimes align with periods of higher forecast errors and compute correlations between average values of forecast errors and the state variable (defined as $y_t - \bar{y}_t$). These are reported in Figure F.7 of the Supplementary Appendix together with a graph of forecast errors and the state variable. The correlation values are 0.93, 0.77, 0.71, and 0.97 for INDPROD, CPIAUCSL, PCEPI,

and UNRATE, respectively. The magnitude of correlations suggests that the large values of the state variable align with periods of large forecast errors.

There are four main findings that we learn from analyzing Table 2: **(1)** For CPIUCSL, PCEPI, and UNRATE RD-FGL significantly outperforms methods without modeling a break. This highlights the importance of defining the regimes using S and $\bar{y}_t$ over the past twelve months. For INDPRO, FGL outperforms RD-FGL. A potential explanation might be the fact that there is less variation in INDPRO compared to other targets, which means there are less benefits of using different regimes. **(2)** For most cases, factor-based models outperform non factor-based counterparts. However, in contrast to the ECB-SPF application in Section 4, the benefits are, on average, less pronounced. We attribute this finding to the difference in the nature of combined forecasts: for ECB SPF we combined the forecasts solicited from experts who use similar information sets to make forecasts and, hence, their errors tend to exhibit common movements especially in the periods of increased uncertainty. Whereas for FRED-MD we created our own forecasts using each macroeconomic series in the dataset as predictors one at a time. Naturally, even though there are still benefits of extracting common components for FRED-MD case, they are weaker compared to combining professional forecasts. **(3)** For all series, the model based on sample covariance provides one of the worst performances. However, its relative performance is better when contrasted with the ECB SPF application. The reason for this improved performance, similarly to our discussion in **(2)**, is somewhat deteriorated performance of the factor-based models. **(4)** Similarly to the ECB SPF application, many combination models still outperform AR(1). However, the relative performance of AR(1) has improved and even belongs to the SSM set.

Table 2: Prediction of Monthly FRED-MD Macroeconomic Variables Using a Combination of Individual Indicators.

	AR1	**GL**	**LW**	**NLW**	**POET**	**CLIME**	**MB**	**Not Sparse**	**FGL**	**FLW**	**FNLW**	**FCLIME**	**FMB**	**RD-FGL (time)**	**RD-FGL (state)**
INDPROD															
$h=1$	1.0155	1.0002	1.2669	1.2534	1.1628	1.0798	0.9971*	3.5701	**0.9884***	0.9972*	1.1317	0.9994	1.0077	0.9907*	1.0600
$h=2$	1.0103	1.0641	1.2012	1.4925	1.3297	1.1057	0.9967	4.1608	**0.9800***	0.9926	1.1615	0.9961	0.9927*	0.9880*	1.0067
CPIAUCSL															
$h=1$	0.9409*	1.0393	1.0503	2.0629	1.1421	1.0453	0.9937	2.0755	1.0013	0.9408*	1.1761	0.9987	0.9862	0.9955	**0.8608***
$h=2$	1.0230	1.0001	1.3364	2.4446	2.8923	1.1545	0.9944	2.8296	0.9928	1.1086	1.6046	0.9972	0.9971	1.0314	**0.9614***
PCEPI															
$h=1$	0.9642*	1.0034	1.1491	6.2377	1.1371	1.0861	0.9971	2.0030	1.0029	0.9646*	1.1428	0.9985	1.0024	1.0445	**0.9307***
$h=2$	1.0659	1.0028	1.3047	2.0436	1.5532	1.0427	0.9981	10.4048	0.9823	1.0186	1.1546	0.9977	0.9950	1.1757	**0.9695***
UNRATE															
$h=1$	0.9972	1.0218	1.0465	2.3550	4.3164	1.0300	1.0045	8.9706	0.9974	1.0075	1.5768	0.9992	1.5814	1.0673	**0.8891***
$h=2$	1.0036	0.9974	1.0096	2.5114	5.5245	1.0083	1.0278	9.3321	0.9929*	0.9928*	1.1457	0.9991	1.0010	1.1658	**0.9897***

Note: MSFEs of competing methods are reported for each value of R, where R indicates the length of the training window. RD-FGL (time) is our method with a time-break, and RD-FGL (state) is our method with a state switch. Ratio indicates the ratio to MSFE of the Equal-Weighted combined forecast. Models with the lowest ratio are in bold. Models marked with a star belong to the SSM according to MCS test which ranks them according to the relative sample loss of the i-th model relative to the average across models in SSM (at 90% confidence level).

6 Conclusions

In this paper we develop a framework to generalize network inference under a factor structure in the presence of structural breaks. We overcome the challenge of using graphical models under the factor structure and provide a simple approach that allows practitioners to combine a large number of forecasts when experts tend to make common errors. Using pre- and post-break data, our new approach to forecast combinations breaks down forecast errors into common and unique parts which improves the accuracy of the combined forecast. We allow the structural breaks to affect factor loadings and idiosyncratic precision matrix. For the ease of practical use we develop a scalable optimization procedure for RD-FGL, based on the ADMM. Two empirical applications to forecasting macroeconomic series using the data of the ECB Survey of Professional Forecasters and FRED-MD show that incorporating (i) factor structure in the forecast errors together with (ii) sparsity in the precision matrix of the idiosyncratic components and (iii) regime-dependent combination weights improves the performance of a combined forecast.

7 Data and Code Availability

The code and data for reproducing the results in the paper are located in the Github repository. We ran empirical results on ml.m5.4xlarge AWS instance (vCPU = 16, memory = 64 GiB), and we ran the simulations on ml.r5.4xlarge AWS instance (vCPU = 16, memory = 128 GiB). Data instructions for ECB SPF and FRED MD data are located in the `ReadMe.txt` file, including the instructions for updating the datasets with most recent data. Our method is written in Python, whereas the competing methods for empirical application are implemented in R since we used the packages written in R by the original authors of

those methods (like CLIME, LW, NLW, POET etc).

References

Bai, J. (2010). Common breaks in means and variances for panel data. *Journal of Econometrics*, 157(1):78–92. 5, 19, 20

Bai, J., Han, X., and Shi, Y. (2020). Estimation and inference of change points in high-dimensional factor models. *Journal of Econometrics*, 219(1):66–100. 5, 19

Bai, J. and Ng, S. (2002). Determining the number of factors in approximate factor models. *Econometrica*, 70(1):191–221. 7, 22, 27

Bai, J. and Perron, P. (2003). Computation and analysis of multiple structural change models. *Journal of applied econometrics*, 18(1):1–22. 20, 24

Bates, J. M. and Granger, C. W. J. (1969). The combination of forecasts. *Operations Research*, 20(4):451–468. 2, 8

Boot, T. and Nibbering, D. (2019). Forecasting using random subspace methods. *Journal of Econometrics*, 209(2):391–406. 22

Cai, T., Liu, W., and Luo, X. (2011). A constrained l1-minimization approach to sparse precision matrix estimation. *Journal of the American Statistical Association*, 106(494):594–607. 22

Chudik, A., Pesaran, M. H., and Sharifvaghefi, M. (2024). Variable selection in high dimensional linear regressions with parameter instability. *Journal of Econometrics*, 246(1-2):105900. 13, 23, 27

Claeskens, G., Magnus, J. R., Vasnev, A. L., and Wang, W. (2016). The forecast combination puzzle: A simple theoretical explanation. *International Journal of Forecasting*, 32(3):754–762. 8

Clemen, R. T. (1989). Combining forecasts: A review and annotated bibliography. *International Journal of Forecasting*, 5(4):559–583. 1

Corradi, V. and Swanson, N. R. (2014). Testing for structural stability of factor augmented forecasting models. *Journal of Econometrics*, 182(1):100–118. Causality, Prediction, and Specification Analysis: Recent Advances and Future Directions. 4, 12

Diebold, F. and Shin, M. (2019). Machine learning for regularized survey forecast combination: Partially-egalitarian lasso and its derivatives. *International Journal of Forecasting*, 35(4):1679–1691. 1, 46

Fan, J., Liao, Y., and Mincheva, M. (2013). Large covariance estimation by thresholding principal orthogonal complements. *Journal of the Royal Statistical Society: Series B*, 75(4):603–680. 22

Fan, J., Liu, H., and Wang, W. (2018). Large covariance estimation through elliptical factor models. *The Annals of Statistics*, 46(4):1383–1414. 11

Friedman, J., Hastie, T., and Tibshirani, R. (2008). Sparse inverse covariance estimation with the Graphical Lasso. *Biostatistics*, 9(3):432–441. 3, 8, 36, 37

Genre, V., Kenny, G., Meyler, A., and Timmermann, A. (2013). Combining expert forecasts: Can anything beat the simple average? *International Journal of Forecasting*, 29(1):108–121. 24

Gibbs, C. G. and Vasnev, A. L. (2024). Conditionally optimal weights and forward-looking approaches to combining forecasts. *International Journal of Forecasting*, 40(4):1734–1751. 8

Hallac, D., Park, Y., Boyd, S., and Leskovec, J. (2017). Network inference via the time-varying graphical lasso. In *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '17, pages 205–213, New York, NY, USA. ACM. 17

Hansen, B. E. (2008). Least-squares forecast averaging. *Journal of Econometrics*, 146(2):342–350. 49

Hansen, P. R., Lunde, A., and Nason, J. M. (2011). The model confidence set. *Econometrica*, 79(2):453–497. 24

Janková, J. and van de Geer, S. (2018). Inference in high-dimensional graphical models. *Handbook of Graphical Models*, Chapter 14, pages 325–351. CRC Press. 12, 37

Ledoit, O. and Wolf, M. (2004). A well-conditioned estimator for large-dimensional covariance matrices. *Journal of Multivariate Analysis*, 88(2):365–411. 22, 47, 53

Ledoit, O. and Wolf, M. (2017). Nonlinear shrinkage of the covariance matrix for portfolio selection: Markowitz meets goldilocks. *The Review of Financial Studies*, 30(12):4349–4388. 22

Lee, T.-H. and Seregina, E. (2024). Optimal Portfolio Using Factor Graphical Lasso. *Journal of Financial Econometrics*, 22(3):670–695. 4, 8, 9, 11, 12

Li, Q., Ouyang, D., and Racine, J. S. (2013). Categorical semiparametric varying-coefficient models. *Journal of Applied Econometrics*, 28(4):551–579. 13

McCracken, M. W. and Ng, S. (2016). FRED-MD: A monthly database for macroeconomic research. *Journal of Business & Economic Statistics*, 34(4):574–589. 1, 26

Meinshausen, N. and Bühlmann, P. (2006). High-dimensional graphs and variable selection with the Lasso. *The Annals of Statistics*, 34(3):1436–1462. 4, 22

Mohan, K., London, P., Fazel, M., Witten, D., and Lee, S.-I. (2014). Node-based learning of multiple gaussian graphical models. *The Journal of Machine Learning Research*, 15(1):445–488. 18

Parikh, N. and Boyd, S. (2014). Proximal algorithms. *Found. Trends Optim.*, 1(3):127–239. 17, 44

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., and Duchesnay, E. (2011). Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830. 47

Pesaran, M. H., Pick, A., and Pranovich, M. (2013). Optimal forecasts in the presence of structural breaks. *Journal of Econometrics*, 177(2):134–152. 13

Radchenko, P., Vasnev, A. L., and Wang, W. (2023). Too similar to combine? on negative weights in forecast combination. *International Journal of Forecasting*, 39(1):18–38. 9

Ravikumar, P., J. Wainwright, M., Raskutti, G., and Yu, B. (2011). High-dimensional covariance estimation by minimizing -penalized log-determinant divergence. *Electronic Journal of Statistics*, 5:935–980. 35

Smith, J. and Wallis, K. F. (2009). A simple explanation of the forecast combination puzzle. *Oxford Bulletin of Economics and Statistics*, 71(3):331–355. 8

Stekhoven, D. J. (2022). *missForest: Nonparametric Missing Value Imputation using Random Forest*. R package version 1.5. 21

Stekhoven, D. J. and Buhlmann, P. (2012). Missforest - non-parametric missing value imputation for mixed-type data. *Bioinformatics*, 28(1):112–118. 21

Stock, J. H. and Watson, M. W. (2002). Forecasting using principal components from a large number of predictors. *Journal of the American Statistical Association*, 97(460):1167–1179. 7, 48

Su, L. and Wang, X. (2017). On time-varying factor models: Estimation and testing. *Journal of Econometrics*, 198(1):84–101. 13, 14, 18

Thompson, R., Qian, Y., and Vasnev, A. L. (2024). Flexible global forecast combinations. *Omega*, 126:103073. 24

Supplemental Appendix to "Combining Forecasts under Structural Breaks Using Graphical LASSO"

Appendix A Graphical Lasso Algorithm

Recall that we have p competing forecasts of the univariate series y_t, $t = 1, \ldots, T$. Let $\mathbf{e}_t = (e_{1t}, \ldots, e_{pt})' \sim \mathcal{N}(\mathbf{0}, \mathbf{\Sigma})$ be a $p \times 1$ vector of forecast errors. Assume they follow a Gaussian distribution. The precision matrix $\mathbf{\Sigma}^{-1} \equiv \mathbf{\Theta}$ contains information about partial covariances between the variables. For instance, if θ_{ij}, which is the ij-th element of the precision matrix, is zero, then the variables i and j are conditionally independent, given the other variables.

Let $\mathbf{W}$ be the estimate of $\mathbf{\Sigma}$. Given a sample $\{\mathbf{e}_t\}_{t=1}^T$, let $\mathbf{S} = (1/T)\sum_{t=1}^T (\mathbf{e}_t)(\mathbf{e}_t)'$ denote the sample covariance matrix, which can be used as a choice for $\mathbf{W}$. Also, let $\widehat{\mathbf{\Gamma}}^2 \equiv \text{diag}(\mathbf{W})$ and its (i,j)-th element is denoted as $\widehat{\gamma}_{ij}$. We can write down truncated Gaussian log-likelihood (up to constants) $l(\mathbf{\Theta}) = \log\det(\mathbf{\Theta}) - \text{tr}(\mathbf{W\Theta})$. When $\mathbf{W} = \mathbf{S}$, the maximum likelihood estimator of $\mathbf{\Theta}$ is $\widehat{\mathbf{\Theta}} = \mathbf{S}^{-1}$. The objective function associated with truncated Gaussian log-likelihood is also known as Bregman divergence and was shown to be applicable for non-Gaussian distributions (Ravikumar et al. (2011)).

In the high-dimensional settings it is necessary to regularize the precision matrix, which means that some edges will be zero. A natural way to induce sparsity in the estimation of precision matrix is to add penalty to the maximum likelihood and use the connection between the precision matrix and regression coefficients to maximize the following penalized log-likelihood that weighs the variables by their scale:

$$\widehat{\mathbf{\Theta}}_\tau = \arg\min_{\mathbf{\Theta}=\mathbf{\Theta}'} \text{tr}(\mathbf{W\Theta}) - \log\det(\mathbf{\Theta}) + \tau \sum_{i \neq j} \widehat{\gamma}_{ii}\widehat{\gamma}_{jj}|\theta_{ij}|, \tag{A.1}$$

over positive definite symmetric matrices, where $\tau \geq 0$ is a penalty parameter for the off-diagonal elements. We refer to the objective function in (A.1) as a "weighted penalized log-likelihood". The subscript τ in $\widehat{\mathbf{\Theta}}_\tau$ means that the solution of the optimization problem in (A.1) will depend upon the choice of the tuning parameter. In order to simplify notation, we will omit the subscript.

Define the following partitions of $\mathbf{W}$, $\mathbf{S}$ and $\boldsymbol{\Theta}$:

$$\mathbf{W} = \begin{pmatrix} \underbrace{\mathbf{W}_{11}}_{(p-1)\times(p-1)} & \underbrace{\mathbf{w}_{12}}_{(p-1)\times 1} \\ \mathbf{w}_{12}' & w_{22} \end{pmatrix}, \mathbf{S} = \begin{pmatrix} \underbrace{\mathbf{S}_{11}}_{(p-1)\times(p-1)} & \underbrace{\mathbf{s}_{12}}_{(p-1)\times 1} \\ \mathbf{s}_{12}' & s_{22} \end{pmatrix}, \boldsymbol{\Theta} = \begin{pmatrix} \underbrace{\boldsymbol{\Theta}_{11}}_{(p-1)\times(p-1)} & \underbrace{\boldsymbol{\theta}_{12}}_{(p-1)\times 1} \\ \boldsymbol{\theta}_{12}' & \theta_{22} \end{pmatrix}. \tag{A.2}$$

Let $\boldsymbol{\beta} \equiv -\boldsymbol{\theta}_{12}/\theta_{22}$. The idea of GL is to set $\mathbf{W} = \mathbf{S} + \tau\mathbf{I}$ in (A.1) and combine the gradient of (A.1) with the formula for partitioned inverses to obtain the following ℓ_1-regularized quadratic program

$$\widehat{\boldsymbol{\beta}} = \arg\min_{\boldsymbol{\beta}\in\mathbb{R}^{p-1}} \left\{ \frac{1}{2}\boldsymbol{\beta}'\mathbf{W}_{11}\boldsymbol{\beta} - \boldsymbol{\beta}'\mathbf{s}_{12} + \sum_{i\neq j} \tau\widehat{\gamma}_{ii}\widehat{\gamma}_{jj}\|\boldsymbol{\beta}\|_1 \right\}. \tag{A.3}$$

As shown by Friedman et al. (2008), (A.3) can be viewed as a LASSO regression, where the LASSO estimates are functions of the inner products of $\mathbf{W}_{11}$ and s_{12}. Hence, (A.1) is equivalent to p coupled LASSO problems. Once we obtain $\widehat{\boldsymbol{\beta}}$, we can estimate the entries $\boldsymbol{\Theta}$ using the formula for partitioned inverses. The weighted GL procedure is summarized in Algorithm A.1.

Algorithm A.1 Weighted Graphical LASSO

1: Initialize $\mathbf{W} = \mathbf{S} + \tau\mathbf{I}$, with $w_{ii} = s_{ii}$. The diagonal of $\mathbf{W}$ remains the same in what follows.

2: Estimate a sparse $\boldsymbol{\Theta}$ using the following weighted Graphical LASSO objective function:

$$\widehat{\boldsymbol{\Theta}}_\tau = \arg\min_{\boldsymbol{\Theta}=\boldsymbol{\Theta}'} \operatorname{tr}(\mathbf{W}\boldsymbol{\Theta}) - \log\det(\boldsymbol{\Theta}) + \tau \sum_{i\neq j} \widehat{\gamma}_{ii}\widehat{\gamma}_{jj}|\theta_{ij}|,$$

over positive definite symmetric matrices.

3: Repeat for $j = 1, \ldots, p, 1, \ldots, p, \ldots$ until convergence:

- Partition $\mathbf{W}$ into part 1: all but the j-th row and column, and part 2: the j-th row and column.
- Solve the score equations using the cyclical coordinate descent:

$$\mathbf{W}_{11}\boldsymbol{\beta} - \mathbf{s}_{12} + \tau\widehat{\gamma}_{ii}\widehat{\gamma}_{jj} \cdot \operatorname{Sign}(\boldsymbol{\beta}) = \mathbf{0}.$$

  This gives a $(p-1)\times 1$ vector solution $\widehat{\boldsymbol{\beta}}$.
- Update $\widehat{\mathbf{w}}_{12} = \mathbf{W}_{11}\widehat{\boldsymbol{\beta}}$.

4: In the final cycle (for $i = 1, \ldots, p$) solve for

$$\frac{1}{\widehat{\theta}_{22}} = w_{22} - \widehat{\boldsymbol{\beta}}'\widehat{\mathbf{w}}_{12}, \quad \widehat{\boldsymbol{\theta}}_{12} = -\widehat{\theta}_{22}\widehat{\boldsymbol{\beta}}.$$

As was shown in Friedman et al. (2008), the estimator produced by Algorithm A.1 is guaranteed to be positive definite. Furthermore, Janková and van de Geer (2018) showed that Algorithm A.1 is guaranteed to converge and produces consistent estimator of precision matrix under certain sparsity conditions.

Appendix B Factor Graphical LASSO

Algorithm B.2 Factor Graphical LASSO (FGL)

1: Estimate factors, $\widehat{\mathbf{f}}_t$, and factor loadings, $\widehat{\mathbf{B}}$, using PCA. Obtain $\widehat{\mathbf{\Sigma}}_f = \frac{1}{T}\sum_{t=1}^{T}\widehat{\mathbf{f}}_t\widehat{\mathbf{f}}_t'$, $\widehat{\mathbf{\Theta}}_f = \widehat{\mathbf{\Sigma}}_f^{-1}$, $\widehat{\boldsymbol{\varepsilon}}_t = \mathbf{e}_t - \widehat{\mathbf{B}}\widehat{\mathbf{f}}_t$, and $\widehat{\mathbf{\Sigma}}_\varepsilon = \frac{1}{T}\sum_{t=1}^{T}\widehat{\boldsymbol{\varepsilon}}_t\widehat{\boldsymbol{\varepsilon}}_t'$.

2: Estimate a sparse $\mathbf{\Theta}_\varepsilon$ using the weighted Graphical LASSO in initialized with $\mathbf{W}_\varepsilon = \widehat{\mathbf{\Sigma}}_\varepsilon + \tau\mathbf{I}$:

$$\widehat{\mathbf{\Theta}}_{\varepsilon,\tau} = \arg\min_{\mathbf{\Theta}_\varepsilon=\mathbf{\Theta}_\varepsilon'} \operatorname{tr}(\mathbf{W}_\varepsilon\mathbf{\Theta}_\varepsilon) - \log\det(\mathbf{\Theta}_\varepsilon) + \tau\sum_{i\neq j}\widehat{\gamma}_{\varepsilon,ii}\widehat{\gamma}_{\varepsilon,jj}|\theta_{\varepsilon,ij}|. \tag{B.1}$$

where $\widehat{\gamma}_{\varepsilon,ii}$ is the (i,i)-th element of $\widehat{\mathbf{\Gamma}}_\varepsilon^2 \equiv \operatorname{diag}(\mathbf{W}_\varepsilon)$.

3: Use $\widehat{\mathbf{\Theta}}_f$ from Step 1 and $\widehat{\mathbf{\Theta}}_\varepsilon$ from Step 2 to estimate $\mathbf{\Theta}$ using the Sherman-Morrison-Woodbury formula:

$$\widehat{\mathbf{\Theta}} = \widehat{\mathbf{\Theta}}_\varepsilon - \widehat{\mathbf{\Theta}}_\varepsilon\widehat{\mathbf{B}}[\widehat{\mathbf{\Theta}}_f + \widehat{\mathbf{B}}'\widehat{\mathbf{\Theta}}_\varepsilon\widehat{\mathbf{B}}]^{-1}\widehat{\mathbf{B}}'\widehat{\mathbf{\Theta}}_\varepsilon. \tag{B.2}$$

Let $\widehat{\mathbf{\Theta}}_{\varepsilon,\tau}$ be the solution to (B.1) for a fixed τ. To choose the optimal shrinkage intensity coefficient, we minimize the following Bayesian Information Criterion (BIC) using grid search:

$$\text{BIC}(\tau) \equiv T\left[\operatorname{tr}(\widehat{\mathbf{\Theta}}_{\varepsilon,\tau}\widehat{\mathbf{\Sigma}}_\varepsilon) - \log\det(\widehat{\mathbf{\Theta}}_{\varepsilon,\tau})\right] + (\log T)\sum_{i\leq j}\mathbb{1}\left[\widehat{\theta}_{\varepsilon,\tau,ij}\neq 0\right]. \tag{B.3}$$

The grid $\mathcal{G} \equiv \{\tau_1,\ldots,\tau_M\}$ is constructed as follows: the maximum value in the grid, τ_M, is set to be the smallest value for which all the off-diagonal entries of $\widehat{\mathbf{\Theta}}_{\varepsilon,\tau_M}$ are zero, that is, the maximum modulus of the off-diagonal entries of $\widehat{\mathbf{\Sigma}}_\varepsilon$. The smallest value of the grid, $\tau_1 \in \mathcal{G}$, is determined as $\tau_1 \equiv \vartheta\tau_M$ for a constant $0 < \vartheta < 1$. The remaining grid values $\tau_1,\ldots,\tau_M$ are constructed in the ascending order from τ_1 to τ_M on the log scale:

$$\tau_i = \exp\Big(\log(\tau_1) + \frac{i-1}{M-1}\log(\tau_M/\tau_1)\Big), \quad i = 2,\ldots,M-1.$$

We use $\vartheta = \sqrt{\log p/T} + 1/\sqrt{p}$ (motivated by the convergence rate from Theorem 1) and $M = 10$ in the simulations and the empirical exercise.

Appendix C Proof of Theorem 1

We first present a lemma which is used in the proof.

Lemma 1.

(a) $|||\mathbf{\Theta}|||_1 = \mathcal{O}(d_T)$.

(b) $a \geq c > 0$, *where* a *was defined in Section 2 and* c *was defined in Assumption* ***(A.1)*** *(ii).*

(c) $|\widehat{a} - a| = \mathcal{O}_P(\varrho_T d_T s_T)$, *where* $\widehat{a}$ *was defined in Section 2.*

Proof.

(a) To prove part (a) we use the following matrix inequality which holds for any $\mathbf{A} \in \mathcal{S}_p$:

$$|||\mathbf{A}|||_1 = |||\mathbf{A}|||_\infty \leq \sqrt{d(\mathbf{A})}|||\mathbf{A}|||_2, \tag{C.1}$$

where $d(\mathbf{A})$ was defined in Section 4. The proof of (C.1) is a straightforward consequence of the Schwarz inequality.

Sherman-Morrison-Woodbury formula together with (C.1) and Assumptions **(B.1)-(B.3)** yield:

$$\begin{aligned} |||\mathbf{\Theta}|||_1 &\leq |||\mathbf{\Theta}_\varepsilon|||_1 + \left|\left|\left|\mathbf{\Theta}_\varepsilon \mathbf{B}[\mathbf{\Theta}_f + \mathbf{B}'\mathbf{\Theta}_\varepsilon \mathbf{B}]^{-1}\mathbf{B}'\mathbf{\Theta}_\varepsilon\right|\right|\right|_1 \\ &= \mathcal{O}(\sqrt{d_T}) + \mathcal{O}\left(\sqrt{d_T} \cdot p \cdot \frac{1}{p} \cdot \sqrt{d_T}\right) = \mathcal{O}(d_T). \end{aligned} \tag{C.2}$$

(b) Under Assumption **(A.1)**:

$$a = \boldsymbol{\iota}_p' \mathbf{\Theta} \boldsymbol{\iota}_p / p \geq c > 0.$$

(c) Using the Hölders inequality, we have

$$|\widehat{a} - a| = \left|\frac{\boldsymbol{\iota}_p'(\widehat{\mathbf{\Theta}} - \mathbf{\Theta})\boldsymbol{\iota}_p}{p}\right| \leq \frac{\left\|(\widehat{\mathbf{\Theta}} - \mathbf{\Theta})\boldsymbol{\iota}_p\right\|_1 \|\boldsymbol{\iota}_p\|_\infty}{p} \leq \left|\left|\left|\widehat{\mathbf{\Theta}} - \mathbf{\Theta}\right|\right|\right|_1$$
$$= \mathcal{O}_P(\varrho_T d_T s_T) = o_P(1),$$

where the last rate is obtained using the assumptions of Theorem 1.

□

C.1 Proof of Theorem 1

First, note that the forecast combination weight can be written as

$$\begin{aligned}\widehat{\mathbf{w}} - \mathbf{w} &= \frac{\Big((a\widehat{\boldsymbol{\Theta}}\boldsymbol{\iota}_p) - (\hat{a}\boldsymbol{\Theta}\boldsymbol{\iota}_p)\Big)/p}{\hat{a}a} \\ &= \frac{\Big((a\widehat{\boldsymbol{\Theta}}\boldsymbol{\iota}_p) - (a\boldsymbol{\Theta}\boldsymbol{\iota}_p) + (a\boldsymbol{\Theta}\boldsymbol{\iota}_p) - (\hat{a}\boldsymbol{\Theta}\boldsymbol{\iota}_p)\Big)/p}{\hat{a}a}.\end{aligned}$$

As shown in Callot et al. (2019), the above can be rewritten as

$$\|\widehat{\mathbf{w}} - \mathbf{w}\|_1 \leq \frac{a\frac{\|(\widehat{\boldsymbol{\Theta}}-\boldsymbol{\Theta})\boldsymbol{\iota}_p\|_1}{p} + |a - \widehat{a}|\frac{\|\boldsymbol{\Theta}\boldsymbol{\iota}_p\|_1}{p}}{|\widehat{a}|a}. \tag{C.3}$$

Prior to bounding the terms in (C.3), we first present an inequality which is used in the derivations. Let $\mathbf{A} \in \mathbb{R}^{p\times p}$ and $\mathbf{v} \in \mathbb{R}^{p\times 1}$. Also, let $\mathbf{A}_j$ and $\mathbf{A}_j'$ be a $p \times 1$ and $1 \times p$ row and column vectors in $\mathbf{A}$, respectively.

$$\begin{aligned}\|\mathbf{A}\mathbf{v}\|_1 &= |\mathbf{A}_1'\mathbf{v}| + \ldots + \left|\mathbf{A}_p'\mathbf{v}\right| \leq \|\mathbf{A}_1\|_1\|\mathbf{v}\|_\infty + \ldots + \|\mathbf{A}_p\|_1\|\mathbf{v}\|_\infty \\ &= \left(\sum_{j=1}^{p}\|\mathbf{A}_j\|_1\right)\|\mathbf{v}\|_\infty \leq p\max_j|\mathbf{A}_j|_1\|\mathbf{v}\|_\infty.\end{aligned} \tag{C.4}$$

Hölders inequality was used to obtain each inequality in (C.4). If $\mathbf{A} \in \mathcal{S}_p$, then the last expression can be further reduced to $p|||\mathbf{A}|||_1\|\mathbf{v}\|_\infty$.

Let us now bound the right-hand side of (C.3). In the numerator we have:

$$\frac{\left\|(\widehat{\boldsymbol{\Theta}} - \boldsymbol{\Theta})\boldsymbol{\iota}_p\right\|_1}{p} \leq |||\boldsymbol{\Theta}|||_1 = \mathcal{O}_P(\varrho_T d_T s_T), \tag{C.5}$$

the rates was derived in Lee and Seregina (2020), and the inequality follows from (C.4).

$$\frac{\|\boldsymbol{\Theta}\boldsymbol{\iota}_p\|_1}{p} \leq |||\boldsymbol{\Theta}|||_1 = \mathcal{O}(d_T), \tag{C.6}$$

where the rate follows from Lemma 1 (a) and the inequality is obtained from (C.4). Com-

bining (C.5), (C.6), and Lemma 1 (c) we get:

$$a\frac{\left\|(\widehat{\boldsymbol{\Theta}}-\boldsymbol{\Theta})\boldsymbol{\iota}_p\right\|_1}{p}+|a-\widehat{a}|\frac{\|\boldsymbol{\Theta}\boldsymbol{\iota}_p\|_1}{p}=\mathcal{O}(1)\cdot\mathcal{O}_P(\varrho_T d_T s_T)+\mathcal{O}_P(\varrho_T d_T s_T)\cdot\mathcal{O}(d_T)$$
$$=\mathcal{O}_P(\varrho_T d_T^2 s_T)=o_P(1), \tag{C.7}$$

where the last equality holds under the assumptions of Theorem 1.

For the denominator of (C.3) it easy to see that $|\widehat{a}|a=\mathcal{O}_P(1)$ using the results of Lemma 1 (b).

For the MSFE part of Theorem 1, using Lemma 1 (b)-(c), we get

$$\left|\frac{\hat{a}^{-1}}{a^{-1}}-1\right|=\frac{|a-\hat{a}|}{|\hat{a}|}=\mathcal{O}_P(\varrho_T d_T s_T)=o_P(1),$$

where the last rate is obtained using the assumptions of Theorem 1.

Appendix D Implementation via ADMM Algorithm

To enable practical implementation of the RD-FGL, we develop an optimization procedure using ADMM algorithm to solve the convex optimization problem in (3.5).

First, we need to reformulate the unconstrained problem in (3.5) as a constrained problem which can be solved using ADMM:

$$\{\widehat{\boldsymbol{\Theta}}_{\varepsilon,j}\}_{j=1}^{N+1} = \arg\min_{\{\boldsymbol{\Theta}_{\varepsilon,j}\}_{j=1}^{N+1}} \sum_{j=1}^{N+1} n_j \Big[\mathrm{tr}\Big(\widehat{\boldsymbol{\Sigma}}_{\varepsilon,j}\boldsymbol{\Theta}_{\varepsilon,j}\Big) - \log\det\boldsymbol{\Theta}_{\varepsilon,j}\Big] + \alpha\|\boldsymbol{\Theta}_{\varepsilon,j}\|_{\mathrm{od},1} \tag{D.1}$$
$$+ \beta \sum_{j=2}^{N+1} \psi(\boldsymbol{\Theta}_{\varepsilon,j} - \boldsymbol{\Theta}_{\varepsilon,j-1})$$

$$\text{s.t. } \mathbf{Z}_{j,0} = \boldsymbol{\Theta}_{\varepsilon,j}, \text{ for } j = 1,\ldots,N+1 \tag{D.2}$$

$$\Big(\mathbf{Z}_{j-1,1}, \mathbf{Z}_{j,2}\Big) = \Big(\boldsymbol{\Theta}_{\varepsilon,j-1}, \boldsymbol{\Theta}_{\varepsilon,j}\Big), \text{ for } j = 2,\ldots,N+1. \tag{D.3}$$

Let $\mathbf{Z} = \Big\{\mathbf{Z}_0, \mathbf{Z}_1, \mathbf{Z}_2\Big\} = \Big\{\Big(\mathbf{Z}_{1,0},\ldots,\mathbf{Z}_{N+1,0}\Big), \Big(\mathbf{Z}_{1,1},\ldots,\mathbf{Z}_{N,1}\Big), \Big(\mathbf{Z}_{2,2},\ldots,\mathbf{Z}_{N+1,2}\Big)\Big\}$.

Let $\mathbf{U} = \Big\{\mathbf{U}_0, \mathbf{U}_1, \mathbf{U}_2\Big\} = \Big\{\Big(\mathbf{U}_{1,0},\ldots,\mathbf{U}_{N+1,0}\Big), \Big(\mathbf{U}_{1,1},\ldots,\mathbf{U}_{N,1}\Big), \Big(\mathbf{U}_{2,2},\ldots,\mathbf{U}_{N,2}\Big)\Big\}$ be the scaled dual variable and $\rho > 0$ is the ADMM penalty parameter. Now we can use scaled ADMM to write down the augmented Lagrangian:

$$\mathcal{L}_\rho(\boldsymbol{\Theta}_\varepsilon, \mathbf{Z}, \mathbf{U}) = \sum_{j=1}^{N+1} n_j \Big[\mathrm{tr}\Big(\widehat{\boldsymbol{\Sigma}}_{\varepsilon,j}\boldsymbol{\Theta}_{\varepsilon,j}\Big) - \log\det\boldsymbol{\Theta}_{\varepsilon,j}\Big] + \alpha\|\mathbf{Z}_{j,0}\|_{od,1} \tag{D.4}$$
$$+ \beta \sum_{j=2}^{N+1} \psi(\mathbf{Z}_{j,2} - \mathbf{Z}_{j-1,1})$$
$$+ \Big(\frac{\rho}{2}\Big) \sum_{j=1}^{N+1} \Big(\|\boldsymbol{\Theta}_{\varepsilon,j} - \mathbf{Z}_{j,0} + \mathbf{U}_{j,0}\|_F^2 - \|\mathbf{U}_{j,0}\|_F^2\Big)$$
$$+ \Big(\frac{\rho}{2}\Big) \sum_{j=2}^{N+1} \Big(\|\boldsymbol{\Theta}_{\varepsilon,j-1} - \mathbf{Z}_{j-1,1} + \mathbf{U}_{j-1,1}\|_F^2 - \frac{\rho}{2}\|\mathbf{U}_{j-1,1}\|_F^2$$
$$+ \|\boldsymbol{\Theta}_{\varepsilon,j} - \mathbf{Z}_{j,2} + \mathbf{U}_{j,2}\|_F^2 - \|\mathbf{U}_{j,2}\|_F^2\Big).$$

Let k denote the iteration number, then ADMM consists of the following iterative updates:

$$\boldsymbol{\Theta}_{\varepsilon,j}^{k+1} \equiv \arg\min_{\boldsymbol{\Theta}_\varepsilon \succ 0} \mathcal{L}_\rho(\boldsymbol{\Theta}_\varepsilon, \mathbf{Z}^k, \mathbf{U}^k), \tag{D.5}$$

$$\mathbf{Z}^{k+1} = \begin{bmatrix} \mathbf{Z}_0^{k+1} \\ \mathbf{Z}_1^{k+1} \\ \mathbf{Z}_2^{k+1} \end{bmatrix} \equiv \arg\min_{\mathbf{Z}_0, \mathbf{Z}_1, \mathbf{Z}_2} \mathcal{L}_\rho(\boldsymbol{\Theta}_\varepsilon^{k+1}, \mathbf{Z}, \mathbf{U}^k), \tag{D.6}$$

$$\mathbf{U}^{k+1} = \begin{bmatrix} \mathbf{U}_0^{k+1} \\ \mathbf{U}_1^{k+1} \\ \mathbf{U}_2^{k+1} \end{bmatrix} \equiv \begin{bmatrix} \mathbf{U}_0^{k} \\ \mathbf{U}_1^{k} \\ \mathbf{U}_2^{k} \end{bmatrix} + \begin{bmatrix} \boldsymbol{\Theta}_\varepsilon^{k+1} - \mathbf{Z}_0^{k+1} \\ (\boldsymbol{\Theta}_{\varepsilon,1}^{k+1}, \ldots, \boldsymbol{\Theta}_{\varepsilon,N}^{k+1}) - \mathbf{Z}_1^{k+1} \\ (\boldsymbol{\Theta}_{\varepsilon,2}^{k+1}, \ldots, \boldsymbol{\Theta}_{\varepsilon,N+1}^{k+1}) - \mathbf{Z}_2^{k+1} \end{bmatrix}. \tag{D.7}$$

The Z step:
The updating rule in (D.6) is easily recognized to be the element-wise soft thresholding operator. However, we need to split it into two updates since $(\mathbf{Z}_1, \mathbf{Z}_2)$ have to be updated jointly. Therefore, the update for $\mathbf{Z}_{j,0}^{k+1}$ will be:

$$\mathbf{Z}_{j,0}^{k+1} \equiv S_{\alpha/\rho}(\boldsymbol{\Theta}_{\varepsilon,j}^{k+1} + \mathbf{U}_{j,0}^k), \tag{D.8}$$

where $S_{\alpha/\rho}(\cdot)$ is the element-wise soft-thresholding operator.

We will solve a separate update for each $(\mathbf{Z}_{j,2}, \mathbf{Z}_{j-1,1})$ pair for $j = 2, \ldots, N+1$:

$$\begin{aligned}(\mathbf{Z}_{j,2}^{k+1}, \mathbf{Z}_{j-1,1}^{k+1}) = \arg\min_{\mathbf{Z}_{j,2}, \mathbf{Z}_{j-1,1}} \left(\frac{\rho}{2}\right) \Big(& \|\boldsymbol{\Theta}_{\varepsilon,j} - \mathbf{Z}_{j,2} + \mathbf{U}_{j,2}\|_F^2 \\ & + \|\boldsymbol{\Theta}_{\varepsilon,j-1} - \mathbf{Z}_{j-1,1} + \mathbf{U}_{j-1,1}\|_F^2 + \beta\psi(\mathbf{Z}_{j,2} - \mathbf{Z}_{j-1,1}) \Big).\end{aligned} \tag{D.9}$$

Note that (D.9) is guaranteed to converge to a fixed point since it can be written as a proximal operator:

$$(\mathbf{Z}_{j,2}^{k+1}, \mathbf{Z}_{j-1,1}^{k+1}) = \text{prox}_{\frac{\beta}{\rho}\psi(\cdot)}\Big(\boldsymbol{\Theta}_{\varepsilon,j} + \mathbf{U}_{j,2}, \boldsymbol{\Theta}_{\varepsilon,j-1} + \mathbf{U}_{j-1,1}\Big) \tag{D.10}$$

Remark 5. *A proximal operator of the scaled function νf, where $\nu > 0$ can be expressed as:*

$$prox_{\nu f}(v) = \underset{x}{\operatorname{argmin}} \left(f(x) + \frac{1}{2\nu} \|x - v\|_2^2 \right),$$

where f is a closed proper convex function. Note:

$$prox_{\nu f}(v) \approx v - \tau \nabla f(v).$$

Parikh and Boyd (2014) show that the fixed points of the proximal operator of f are precisely the minimizers of f, i.e., $prox_{\nu f}(x^\star) = x^\star$ if and only if $x^\star$ minimizes f.

The $\boldsymbol{\Theta}$ step:
The updating rule in (D.5) can be further simplified to obtain a closed-form solution. Rewrite (D.5):

$$\boldsymbol{\Theta}_{\varepsilon,j}^{k+1} = \underset{\Theta_\varepsilon \succ 0}{\operatorname{argmin}} \operatorname{tr}\left(\widehat{\boldsymbol{\Sigma}}_j \boldsymbol{\Theta}_{\varepsilon,j} \right) - \log\det \boldsymbol{\Theta}_{\varepsilon,j} + \frac{1}{2\eta} \left\| \boldsymbol{\Theta}_{\varepsilon,j} - \mathbf{A}^k \right\|_F^2, \tag{D.11}$$

where $\mathbf{A}^k = \dfrac{\mathbf{Z}_{i,0}^k + \mathbf{Z}_{j-1,1}^k + \mathbf{Z}_{j,2}^k - \mathbf{U}_{j,0}^k - \mathbf{U}_{j-1,1}^k - \mathbf{U}_{j,2}^k}{3}$, and $\eta = \frac{n_j}{3\rho}$.

Take the gradient of the updating rule in (D.11) in order to get an analytical solution:

$$\widehat{\boldsymbol{\Sigma}}_{\varepsilon,j} - \boldsymbol{\Theta}_{\varepsilon,j}^{-1,(k+1)} + \frac{1}{\eta}\left(\boldsymbol{\Theta}_{\varepsilon,j}^{k+1} - \mathbf{A}^k \right) = 0, \tag{D.12}$$

$$\frac{1}{\eta}\boldsymbol{\Theta}_{\varepsilon,j}^{k+1} - \boldsymbol{\Theta}_{\varepsilon}^{-1,(k+1)} = \frac{1}{\eta}\mathbf{A}^k - \widehat{\boldsymbol{\Sigma}}_{\varepsilon,j}. \tag{D.13}$$

Equation (D.13) implies that $\boldsymbol{\Theta}_{\varepsilon,j}^{k+1}$ and $\frac{1}{\eta}\mathbf{A}^k - \widehat{\boldsymbol{\Sigma}}_{\varepsilon,j}$ share the same eigenvectors.

Let $\mathbf{Q}_j \boldsymbol{\Lambda}_j \mathbf{Q}_j'$ be the eigendecomposition of $\frac{1}{\eta}\mathbf{A}^k - \widehat{\boldsymbol{\Sigma}}_{\varepsilon,j}$, where $\boldsymbol{\Lambda}_j = \operatorname{diag}(\lambda_{1,j}, \ldots, \lambda_{p,j})$, and $\mathbf{Q}_j'\mathbf{Q}_j = \mathbf{Q}_j\mathbf{Q}_j' = \mathbf{I}$.[7] Pre-multiply (D.13) by $\mathbf{Q}_j'$ and post-multiply it by $\mathbf{Q}_j$:

$$\frac{1}{\eta}\widetilde{\boldsymbol{\Theta}}_{\varepsilon,j}^{k+1} - \widetilde{\boldsymbol{\Theta}}_{\varepsilon,j}^{-1,(k+1)} = \boldsymbol{\Lambda}_j. \tag{D.14}$$

[7] Note that in practice we need to check that $\mathbf{A}^k$ is symmetric. If it is not, then we can define $\widetilde{\mathbf{A}} \equiv \frac{\mathbf{A}^k + (\mathbf{A}^k)'}{2}$ and use it in the described algorithm instead of $\mathbf{A}^k$. Since $\boldsymbol{\Theta}_j$ is symmetric the results will not be affected.

Now construct a diagonal solution of (D.14):

$$\frac{1}{\eta}\tilde{v}_{i,j} - \frac{1}{\tilde{v}_{i,j}} = \lambda_{i,j}, \tag{D.15}$$

where $\tilde{v}_{i,j}$ denotes the i-th eigenvalue of $\widetilde{\mathbf{\Theta}}_{\varepsilon,j}$. Solving for $\tilde{v}_{i,j}$ we get:

$$\tilde{v}_{i,j} = \frac{\lambda_{i,j} + \sqrt{\lambda_{i,j}^2 + \frac{4}{\eta}}}{2\eta^{-1}}. \tag{D.16}$$

Now we can calculate $\mathbf{\Theta}_{\varepsilon,j}^{k+1}$ which satisfies the optimality condition in (D.14):

$$\mathbf{\Theta}_{\varepsilon,j}^{k+1} = \frac{1}{2\eta^{-1}}\mathbf{Q}_j\Big(\mathbf{\Lambda}_j + \sqrt{\mathbf{\Lambda}_j^2 + 4\eta^{-1}\mathbf{I}}\Big)\mathbf{Q}_j'. \tag{D.17}$$

Use the definition of $\eta = \frac{n_j}{3\rho}$:

$$\mathbf{\Theta}_{\varepsilon,j}^{k+1} = \frac{n_j}{6\rho}\mathbf{Q}_j\Big(\mathbf{\Lambda}_j + \sqrt{\mathbf{\Lambda}_j^2 + \frac{12\rho}{n_j}\mathbf{I}}\Big)\mathbf{Q}_j'. \tag{D.18}$$

Step (D.18) is the most computationally intensive task in the algorithm since the runtime of decomposing a $p \times p$ matrix is $\mathcal{O}(p^3)$. Also, note that compared to standard ADMM without smoothing penalty β, (D.18) enforces stronger shrinkage. This is consistent with our motivation for the additional constraint - to smooth the estimator of precision matrix.

Appendix E Monte Carlo

We divide the simulation results into two families of DGPs. The first one studies the consistency of the FGL and RD-FGL for estimating precision matrix and the combination weights. The second one evaluates the out-of-sample forecasting performance of combined forecasts in terms of MSFE. We compare the performance of forecast combinations based on the factor models in Algorithm 1 (RD-FGL) and Algorithm B.2 (FGL) with equal-weighted (EW) forecast combination, and combinations that use GL without factor structure (Algorithm A.1).[8] We examine the performance of RD-FGL for different specifications of the smoothing function $\psi(\cdot)$ as described in Section 3. LASSO penalty is denoted as ℓ_1, Group LASSO as ℓ_g, and Ridge as ℓ_2. Similarly to the literature on graphical models, all exercises use 100 Monte Carlo simulations. To check robustness of RD-FGL, we present simulation results for several cases: (i) with a structural break in both $\mathbf{B}$ and $\mathbf{\Theta}_\varepsilon$, (ii) without a break, (iii) with break only in $\mathbf{\Theta}_\varepsilon$, and (iv) with multiple breaks.

E.1 Consistent Estimation of Forecast Combination Weights

We consider sparse Gaussian graphical models which may be fully specified by a precision matrix $\mathbf{\Theta}_0$. Therefore, the random sample is distributed as $\mathbf{e}_t = (e_{1t}, \ldots, e_{pt})' \sim \mathcal{N}(0, \mathbf{\Sigma}_0)$, where $\mathbf{\Theta}_0 = (\mathbf{\Sigma}_0)^{-1}$ for $t = 1, \ldots, T,\ i = 1, \ldots, p$. Let $\widehat{\mathbf{\Theta}}$ be the precision matrix estimator. We show consistency of the FGL in (i) the operator norm, $\left|\left|\left|\widehat{\mathbf{\Theta}} - \mathbf{\Theta}_0\right|\right|\right|_2$, and (ii) in ℓ_1-vector norm for the combination weights, $\|\widehat{\mathbf{w}} - \mathbf{w}\|_1$, where $\mathbf{w}$ is given by (2.3).

The forecast errors are assumed to have the following structure:

$$\underbrace{\mathbf{e}_t}_{p\times 1} = \mathbf{B}\underbrace{\mathbf{f}_t}_{q\times 1} + \boldsymbol{\varepsilon}_t,\ \mathbf{f}_t = \phi_f \mathbf{f}_{t-1} + \boldsymbol{\zeta}_t, \quad t = 1, \ldots, T \tag{E.1}$$

where $\mathbf{e}_t$ is a $p \times 1$ vector of forecast errors following $\mathcal{N}(\mathbf{0}, \mathbf{\Sigma})$, $\mathbf{f}_t$ is a $q \times 1$ vector of factors, $\mathbf{B}$ is a $p \times q$ matrix of factor loadings, ϕ_f is an autoregressive parameter in the factors which is a scalar for simplicity, $\boldsymbol{\zeta}_t$ is a $q \times 1$ random vector with each component independently following $\mathcal{N}(0, \sigma_\zeta^2)$, $\boldsymbol{\varepsilon}_t$ is a $p \times 1$ random vector following $\mathcal{N}(0, \mathbf{\Sigma}_\varepsilon)$, with sparse $\mathbf{\Theta}_\varepsilon$ that has a random graph structure described below. To create $\mathbf{B}$ in (E.1) we take the first q columns of an upper triangular matrix from a Cholesky decomposition of the $p \times p$ Toeplitz matrix parameterized by ρ: that is, $\mathbf{B} = (b)_{lm}$, where $(b)_{lm} = \rho^{|l-m|}$, $l, m \in \{1, \ldots, p\}$. We set $\rho = 0.2$, $\phi_f = 0.2$ and $\sigma_\zeta^2 = 1$. The specification in (E.1) leads to the low-rank plus sparse

[8] EW arises when the forecast errors follow a factor structure (one factor, homogeneous idiosyncratic variance). It can be viewed as one of "factor-based" methods. To this extent, egalitarian LASSO of Diebold and Shin (2019) is a special case of FGL.

decomposition of the covariance matrix $\mathbb{E}[\mathbf{e}_t\mathbf{e}_t'] = \mathbf{\Sigma} = \mathbf{B}\mathbf{\Sigma}_f\mathbf{B}' + \mathbf{\Sigma}_\varepsilon$. When $\mathbf{\Sigma}_\varepsilon$ has a sparse inverse $\mathbf{\Theta}_\varepsilon$, it leads to the low-rank plus sparse decomposition of the precision matrix $\mathbf{\Theta}$, such that $\mathbf{\Theta}$ can be expressed as a function of the low-rank $\mathbf{\Theta}_f$ plus sparse $\mathbf{\Theta}_\varepsilon$.

We consider the following setup: let $p = T^\delta$, $\delta = 0.85$, $q = 2(\log(T))^{0.5}$ and $T = [2^\kappa]$, for $\kappa = 7, 7.5, 8, \ldots, 9.5$. Our setup allows the number of individual forecasts, p, and the number of common factors in the forecast errors, q, to increase with the sample size, T.

A sparse precision matrix of the idiosyncratic components $\mathbf{\Theta}_\varepsilon$ is constructed as follows: we first generate the adjacency matrix using a random graph structure. Define a $p \times p$ adjacency matrix $\mathbf{A}_\varepsilon$ which represents the structure of the graph with $a_{\varepsilon,lm}$ being the l, m-th element of the adjacency matrix $\mathbf{A}_\varepsilon$. We set $a_{\varepsilon,lm} = a_{\varepsilon,ml} = 1$, for $l \neq m$ with probability π, and 0 otherwise. Such structure results in $s_T = p(p-1)\pi/2$ edges in the graph. To control sparsity, we set $\pi = 500/(pT^{0.8})$, which makes $s_T = \mathcal{O}(T^{0.05})$. The adjacency matrix has all diagonal elements equal to zero. To generate a sparse symmetric positive-definite precision matrix we use Scikit-Learn datasets package in Python (Pedregosa et al. (2011)). To control the magnitude of partial correlations, the value of the smallest coefficient is set to 0.1 and the value of the largest coefficient is set to 0.3.

To incorporate structural breaks in $\mathbf{\Theta}_\varepsilon$ and factor loadings $\mathbf{B}$, we proceed as follows. We fix a single break point in the middle of the sample size, $T/2$: in the precision matrix of the idiosyncratic errors before the break, referred to as $\mathbf{\Theta}_{\varepsilon,1}$, the value of the largest coefficient is set to 0.4; whereas in the precision matrix of the idiosyncratic errors after the break, $\mathbf{\Theta}_{\varepsilon,2}$, the value of the largest coefficient is set to 0.6. As a consequence, even though both matrices are still sparse, $\mathbf{\Theta}_{\varepsilon,2}$ has larger partial correlations. We use $\mathbf{\Theta}_{\varepsilon,1}$ and $\mathbf{\Theta}_{\varepsilon,2}$ to generate $\boldsymbol{\varepsilon}_t$ in (E.1). For the structural break in factor loadings (which is assumed to happen at the same time as the structural change in $\mathbf{\Theta}_\varepsilon$), before the break we set $\rho_1 = 0.2$ in the Toeplitz matrix used to generate $\mathbf{B}$ (i.e., $\mathbf{B} = (b)_{lm}$, where $(b)_{lm} = \rho^{|l-m|}$), and after the break we set $\rho_2 = 0.6$.

Figure E.1 shows the averaged (over Monte Carlo simulations) errors of the estimators of the precision matrix $\mathbf{\Theta}$ and the optimal combination weight versus the sample size T in the logarithmic scale (base 2). The estimate of the precision matrix of the EW forecast combination is obtained using the fact that diagonal covariance and precision matrices imply equal weights. To determine the values of the diagonal elements we use the shrinkage intensity coefficient calculated as the average of the eigenvalues of the sample covariance matrix of the forecast errors (see Ledoit and Wolf (2004)).

Figure E.1 examines the performance when there are breaks in both $\mathbf{\Theta}_\varepsilon$ and $\mathbf{B}$: accounting for the break significantly reduces the estimation error of precision matrix and combination weights. We report the results for the case when γ is estimated using cross-

validation ($\gamma = \hat{\gamma}$) (as discussed in Section 4). Online Supplement S2 presents the results for the case when the break is only in $\mathbf{\Theta}_\varepsilon$.

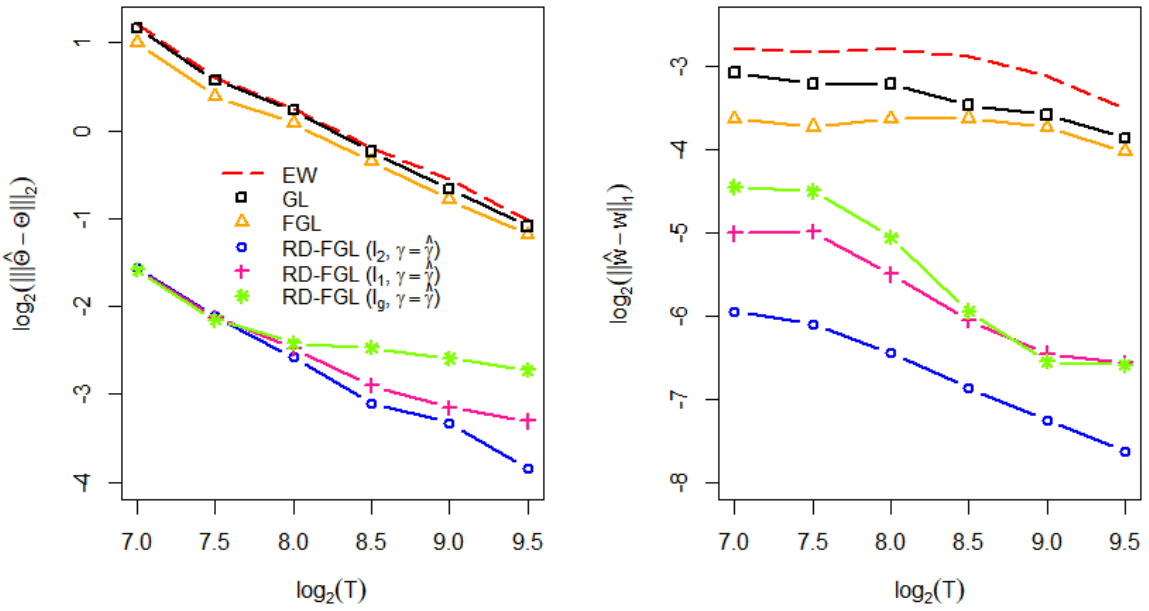

Figure E.1: Averaged errors of the estimators of $\mathbf{\Theta}$ (left) and $\mathbf{w}$ on logarithmic scale (base 2): break in $\mathbf{\Theta}_\varepsilon$ and factor loadings $\mathbf{B}$. $p = T^{0.85}$, $q = 2(\log(T))^{0.5}$, $s_T = \mathcal{O}(T^{0.05})$. The horizontal axis ranges from $T = 2^7, \ldots, T = 2^{9.5}$.

E.2 Comparing Performance of Forecast Combinations

We consider the standard forecasting model in the literature (e.g., Stock and Watson (2002)), which uses the factor structure of the high dimensional predictors. Suppose the data is generated from the following data generating process (DGP):

$$\mathbf{x}_t = \mathbf{\Lambda}\mathbf{g}_t + \mathbf{v}_t, \ \mathbf{g}_t = \phi\mathbf{g}_{t-1} + \boldsymbol{\xi}_t, \ y_{t+1} = \mathbf{g}_t'\boldsymbol{\alpha} + \sum_{s=1}^{\infty}\theta_s\epsilon_{t+1-s} + \epsilon_{t+1}, \tag{E.2}$$

where y_{t+1} is a univariate series of our interest in forecasting, $\mathbf{x}_t$ is an $M \times 1$ vector of regressors (predictors), $\boldsymbol{\alpha}$ is an $M \times 1$ parameter vector, $\mathbf{g}_t$ is an $r \times 1$ vector of factors, $\mathbf{\Lambda}$ is an $M \times r$ matrix of factor loadings, $\mathbf{v}_t$ is an $M \times 1$ random vector following $\mathcal{N}(0, \sigma_v^2\mathbf{I}_M)$, ϕ is an autoregressive parameter in the factors which is a scalar for simplicity, $\boldsymbol{\xi}_t$ is an $M \times 1$ random vector with each component independently following $\mathcal{N}(0, \sigma_\xi^2)$, ϵ_{t+1} is a random

error following $\mathcal{N}(0, \sigma_\epsilon^2)$, and $\boldsymbol{\alpha}$ is an $r \times 1$ parameter vector which is drawn randomly from $\mathcal{N}(1, 1)$. We set $\sigma_\epsilon = 1$. The coefficients θ_s are set according to the rule $\theta_s = (1+s)^{c_1} c_2^s$ as in Hansen (2008). We set $c_1 = 0.75$. We set $M = 100$ and generate $r = 5$ factors. To create $\boldsymbol{\Lambda}$ in (E.2) we take the first r rows of an upper triangular matrix from a Cholesky decomposition of the $M \times M$ Toeplitz matrix parameterized by $\rho = 0.9$. The ranking of competing models was not very sensitive to varying values of ϕ, ρ, c_2, and r.

One-step ahead forecasts are estimated from the factor-augmented autoregressive (FAR) models of orders k, l, denoted as FAR(k, l):

$$\hat{y}_{t+1} = \hat{\mu} + \hat{\kappa}_1 \hat{g}_{1,t} + \cdots + \hat{\kappa}_k \hat{g}_{k,t} + \hat{\psi}_1 y_t + \cdots + \hat{\psi}_l y_{t+1-l}, \tag{E.3}$$

where the factors $(\hat{g}_{1,t}, \ldots, \hat{g}_{k,t})$ are estimated from equation (E.2). We consider the FAR models of various orders, with $k = 1, \ldots, K$ and $l = 1, \ldots, L$. We also consider the models without any lagged y or any factors. Therefore, the total number of forecasting models is $p \equiv (1+K) \times (1+L)$. We set $K = 2$ and $L = 7$.

The total number of observations is m. The period for training the models is set to be $m_1 = m/2$ – this is used to train competing FAR models in (E.3). The remaining part of the sample, $m_2 = m - m_1$ is split as follows: the estimation window for training competing models (that is, EW, GL, FGL, and RD-FGL) is set to be of size $= m_2/2$. We roll the estimation window over the the test sample of the size $m_2/2$ to update all the estimates in each point of time. Recall that q denotes the number of factors in the forecast errors as in equation (2.1).

To incorporate structural break we proceed as follows. The period for training the models is set to be $m_1 = m/3$ – this is used to train competing FAR models in (E.3). The remaining part of the sample, $m_2 = m - m_1$ is split as follows: the estimation window for training competing models is set to be of size $= m_2/2$. We roll the estimation window over the test sample of the size $m_2/2$. The break point is fixed at $1/2$ of the first estimation window. Before the break, when generating θ_s we set $c_2 = 0.3$, and after the break $c_2 = 0.9$. All other parameters stay unchanged. Notice that the break in c_2 can propagate into both a break in precision matrix and factor loadings.

Similarly to the previous subsection, we include different specifications of the smoothing function $\psi(\cdot)$. Figure E.2 shows the performance of all models including RD-FGL with γ estimated using cross-validation: similarly to the conclusions in the previous subsection, accounting for the break significantly reduces MSFE of the combined forecast.

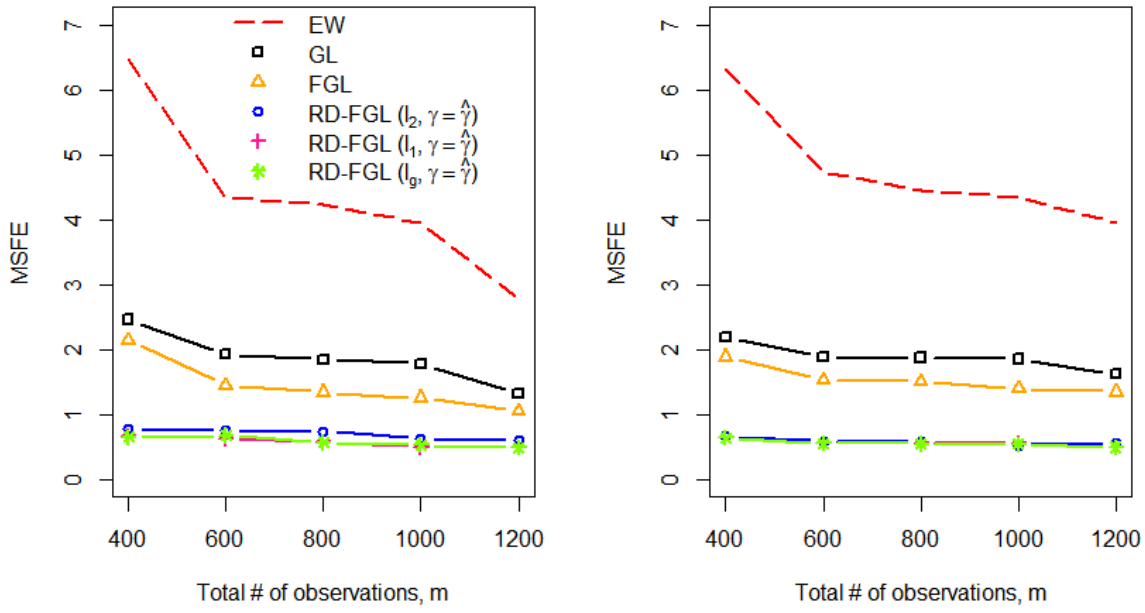

Figure E.2: Plots of the MSFE over the total number of observations m. $c_1 = 0.75$, $c_2 = 0.3$ (before the break), $c_2 = 0.9$ (after the break), $M = 100$, $r = 5, \sigma_\xi = 1$, $L = 7$, $K = 2$, $p = 24$, $q = 3, \phi = 0.8$. Left: $\rho = 0.2$, right: $\rho = 0.9$.

E.3 No Break

In this subsection we present simulation results that augment the results in Section 5 by assuming there is no break in the DGP.

First, we explore behavior of precision matrix and weights estimates. The setup is the same as in Subsection E.1, but there is no break in $\mathbf{\Theta}_\varepsilon$: the value of the smallest coefficient is set to 0.1 and the value of the largest coefficient is set to 0.3.

Figure E.3 shows the averaged (over Monte Carlo simulations) errors of the estimators of the precision matrix $\mathbf{\Theta}$ and the optimal combination weight versus the sample size T in the logarithmic scale (base 2). For comparison, we include RD-FGL in all simulations. Since there is no break in the DGP, the tuning parameter for the factor loadings $\gamma = 1$ and the value of β is estimated to be zero. Henceforth, all specifications of the smoothing function $\psi(\cdot)$ yield similar results and we only include one of them RD-FGL (ℓ_2). As evidenced by Figure E.3, FGL and RD-FGL demonstrate superior performance over EW and non-factor based model (GL). FGL and RD-FGL have comparable performance, but since there is no break in DGP, FGL is more efficient. Furthermore, FGL and RD-FGL achieve lower estimation

error in the combination weights, which leads to lower risk of the combined forecast. Also, note that the precision matrix estimated using the EW method also shows good convergence properties.

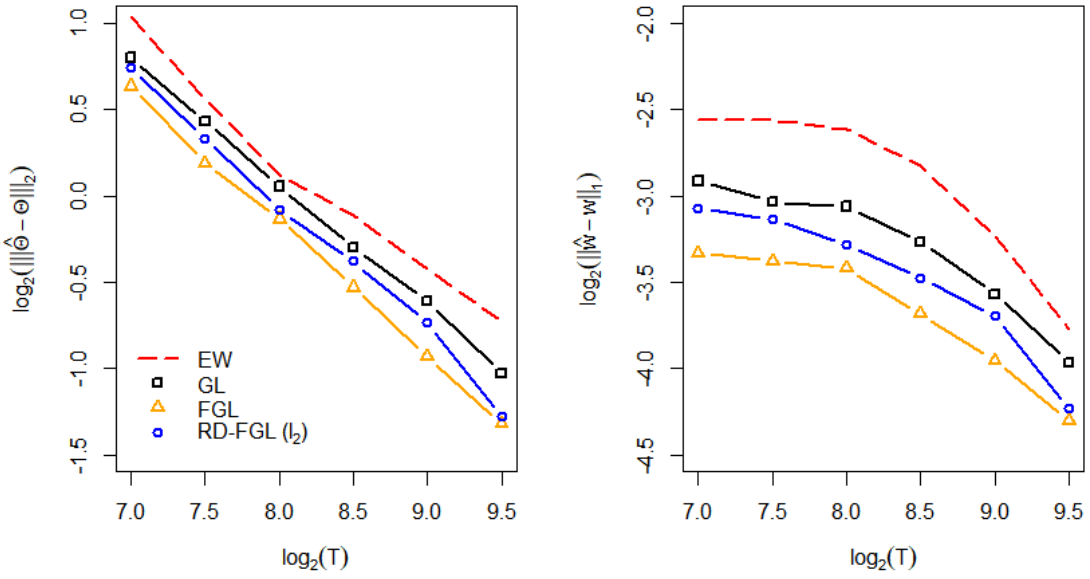

Figure E.3: **Averaged errors of the estimators of $\mathbf{\Theta}$ (left) and $\mathbf{w}$ on logarithmic scale (base 2).** $p = T^{0.85}$, $q = 2(\log(T))^{0.5}$, $s_T = \mathcal{O}(T^{0.05})$. The horizontal axis ranges from $T = 2^7, \ldots, T = 2^{9.5}$.

Second, we explore behavior of MSFE under no breaks. We set $c_1 \in \{0, 0.75\}$ and $c_2 = 0.9$. We set $M = 100$ and generate $r = 5$ factors. To create $\mathbf{\Lambda}$ in (E.2) we take the first r rows of an upper triangular matrix from a Cholesky decomposition of the $M \times M$ Toeplitz matrix parameterized by $\rho = 0.9$. The ranking of competing models was not very sensitive to varying values of ϕ, ρ, c_2, and r – the results examining sensitivity to a grid of 10 different AR(1) coefficients ϕ equidistant between 0 and 0.9, a grid of 10 different values of ρ equidistant between 0 and 0.9, $c_2 \in \{0.6, 0.7, 0.8, 0.9\}$, and $r \in \{1, \cdots, 7\}$ are available upon request.

One-step ahead forecasts are estimated from the factor-augmented autoregressive (FAR) models of orders k, l, denoted as FAR(k, l), defined in (E.3). We consider the FAR models of various orders, with $k = 1, \ldots, K$ and $l = 1, \ldots, L$. We also consider the models without any lagged y or any factors. Therefore, the total number of forecasting models is $p \equiv (1 + K) \times (1 + L)$, which includes the forecasting models using naive average or no factors. We set $K = 2$ and $L = 7$.

The total number of observations is m. The period for training the models is set to be $m_1 = m/2$ – this is used to train competing FAR models in (E.3). The remaining part of the sample, $m_2 = m - m_1$ is split as follows: the estimation window for training competing models (that is, EW, GL, FGL, and RD-FGL) is set to be window $= m_2/2$. We roll the estimation window over the the test sample of the size $m_2/2$ to update all the estimates in each point of time. Recall that q denotes the number of factors in the forecast errors as in equation (2.1).

Similarly to the previous subsection, we include RD-FGL in all simulations. When there is no break in the DGP, the tuning parameter for the factor loadings, γ, is set to one, and the penalty that controls the change of idiosyncratic precision matrix over time, β, is zero. Figure E.4 shows the MSFE for different sample sizes and fixed parameters: we report the results for two values of $c_1 \in \{0, 0.75\}$. As evidenced from Figure E.4, the models that use the factor structure outperform EW combination and non-factor based counterparts for both values of c_1.

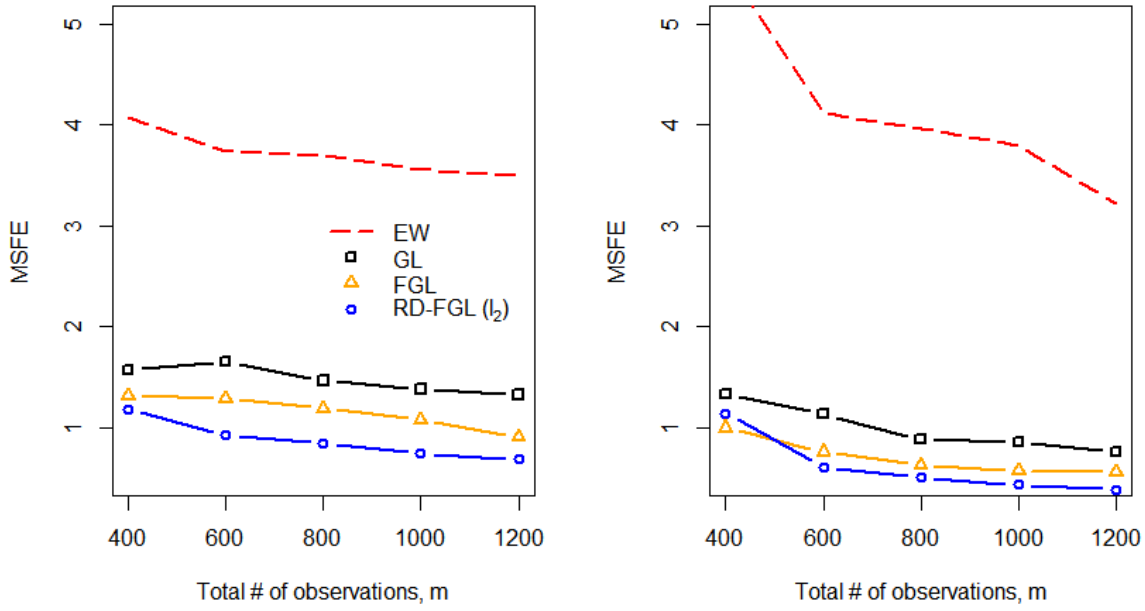

Figure E.4: **Plots of the MSFE over the total number of observations m.** $c_1 = 0$ (left), $c_1 = 0.75$ (right), $c_2 = 0.9$, $M = 100$, $r = 5, \sigma_\xi = 1$, $L = 7$, $K = 2$, $p = 24$, $q = 3$, $\rho = 0.9$, $\phi = 0.8$.

E.4 Break Only in Idiosyncratic Precision Matrix

This subsection presents the results for the case when there is a single break in $\mathbf{\Theta}_\varepsilon$. The DGP is the same as described in Subsection 5.1: the break point is fixed in the middle of the sample $T/2$. Before the break, the value of the largest coefficient in $\mathbf{\Theta}_{\varepsilon,1}$ is set to 0.4, after the break it changes to 0.6.

Figure E.5 shows the averaged (over Monte Carlo simulations) errors of the estimators of the precision matrix $\mathbf{\Theta}$ and the optimal combination weight versus the sample size T in the logarithmic scale (base 2). The estimate of the precision matrix of the EW forecast combination is obtained using the fact that diagonal covariance and precision matrices imply equal weights. To determine the values of the diagonal elements we use the shrinkage intensity coefficient calculated as the average of the eigenvalues of the sample covariance matrix of the forecast errors (see Ledoit and Wolf (2004)).

Figure E.5 shows the performance of all models including RD-FGL when the break is only in $\mathbf{\Theta}_\varepsilon$ ($\gamma = 1$): accounting for the break significantly reduces the estimation error of precision matrix and combination weights.

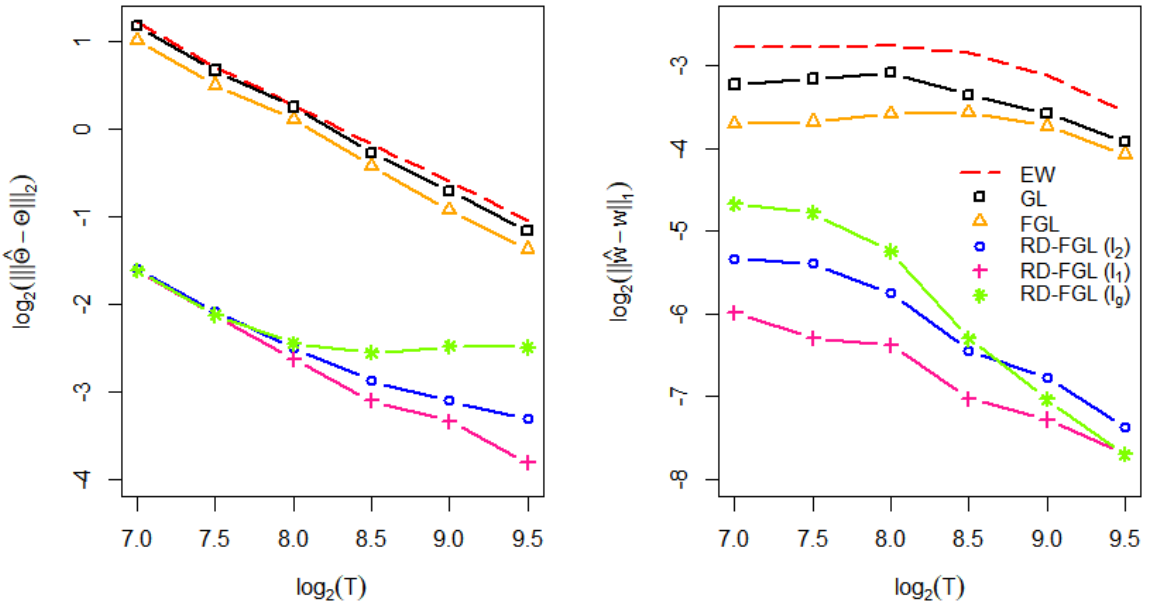

Figure E.5: **Averaged errors of the estimators of $\mathbf{\Theta}$ (left) and $\mathbf{w}$ on logarithmic scale (base 2): break in $\mathbf{\Theta}_\varepsilon$.** $p = T^{0.85}$, $q = 2(\log(T))^{0.5}$, $s_T = \mathcal{O}(T^{0.05})$. The horizontal axis ranges from $T = 2^7, \ldots, T = 2^{9.5}$.

E.5 Multiple Breaks

We examine the performance of RD-FGL and competing methods for the case of two known breaks.

First, we explore behavior of precision matrix and weights estimates. To incorporate two structural breaks in $\mathbf{\Theta}_\varepsilon$, we add the following modification to the DGP setup in Subsection 5.1. We fix two break points: one at $t_1 = T/4$ and the other at $t_1 = 3T/4$. Define the following idiosyncratic precision matrices: $\mathbf{\Theta}_{\varepsilon,1}$ before t_1, $\mathbf{\Theta}_{\varepsilon,2}$ between t_1 and t_2, and $\mathbf{\Theta}_{\varepsilon,3}$ after t_2. The value of the largest coefficient in the three aforementioned matrices is set to 0.2, 0.4, and 0.6, accordingly.

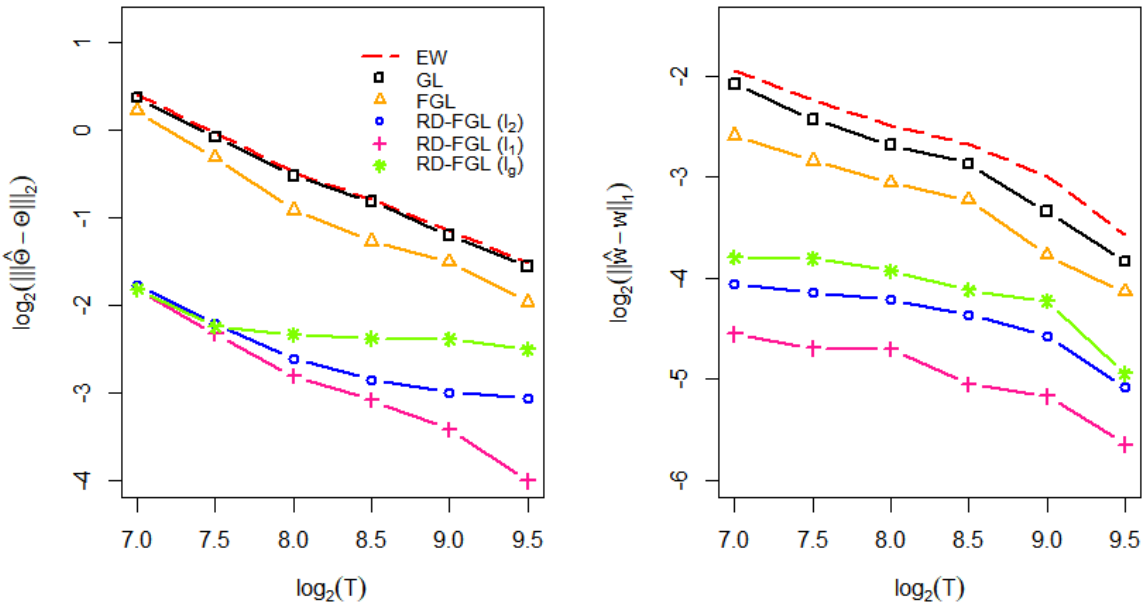

Figure E.6: **Averaged errors of the estimators of $\mathbf{\Theta}$ (left) and w on logarithmic scale (base 2): two breaks in $\mathbf{\Theta}_\varepsilon$.** $p = T^{0.85}$, $q = 2(\log(T))^{0.5}$, $s_T = \mathcal{O}(T^{0.05})$. The horizontal axis ranges from $T = 2^7, \ldots, T = 2^{9.5}$.

As demonstrated in Figure E.6, similarly to the findings in the main manuscript for the case with one break, accounting for the break significantly reduces the estimation error of precision matrix and combination weights.

Second, we explore behavior of MSFE under two breaks. To incorporate two structural breaks we add the following modification to the DGP in Subsection E.2. The total number of observations is m. The period for training the models is set to be $m_1 = T/3$ – this is used

to train competing FAR models in (E.3). The remaining part of the sample, $m_2 = m - m_1$ is split similarly to Subsection E.2: the estimation window for training competing models is set to be window $= m_2/2$. We roll the estimation window over the the test sample. The break points are fixed at 1/3 and 3/4 of the first estimation window, and will be referred to as t_1 and t_2.

When generating θ_s we set c_2 as follows: $c_2 = 0.3$ before t_1, $c_2 = 0.6$ between t_1 and t_2, $c_3 = 0.9$ after t_2.

Before the break, when generating θ_s we set $c_2 = 0.3$, and after the break $c_2 = 0.9$. All other parameters stay unchanged. Notice that the break in c_2 can propagate into both a break in precision matrix and factor loadings.

Similarly to the main manuscript, we include different specifications of the smoothing function $\psi(\cdot)$. Figure E.7 shows the performance of all models including RD-FGL with γ estimated using cross-validation: similarly to the conclusions in Subsection E.2, accounting for the break significantly reduces MSFE of the combined forecast.

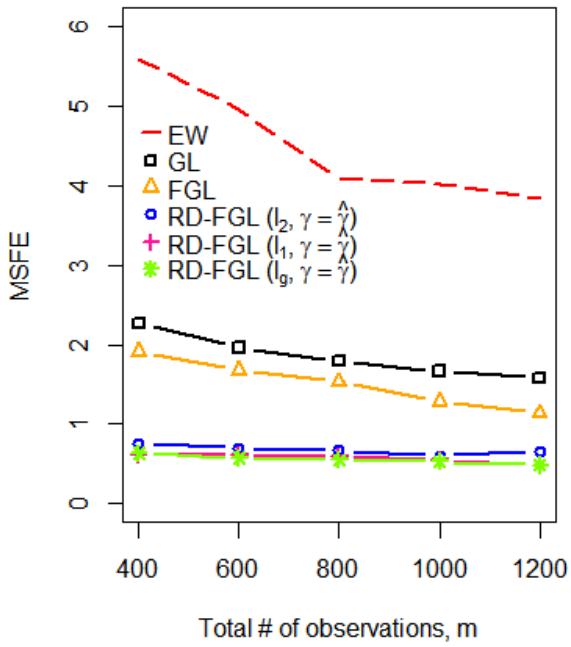

Figure E.7: **Plots of the MSFE over the total number of observations m.** $c_1 = 0.75$, $c_2 = 0.3$ (before t_1), $c_2 = 0.6$ (between t_1 and t_2), $c_3 = 0.9$ (after t_2), $M = 100$, $r = 5, \sigma_\xi = 1$, $L = 7$, $K = 2$, $p = 24$, $q = 3$, $\rho = 0.9$, $\phi = 0.8$.

E.6 Varying Break Magnitude

We examine the performance of RD-FGL and competing methods for the case of one known break of smaller magnitude.

First, we explore behavior of precision matrix and weights estimates. The setup is the same as in Subsection E.1: we fix a single break point in the middle of the sample size, $T/2$: in the precision matrix of the idiosyncratic errors before the break, referred to as $\mathbf{\Theta}_{\varepsilon,1}$, the value of the largest coefficient is set to 0.4; whereas in the precision matrix of the idiosyncratic errors after the break, $\mathbf{\Theta}_{\varepsilon,2}$, the value of the largest coefficient is set to 0.45. We use $\mathbf{\Theta}_{\varepsilon,1}$ and $\mathbf{\Theta}_{\varepsilon,2}$ to generate $\boldsymbol{\varepsilon}_t$ in (E.1).

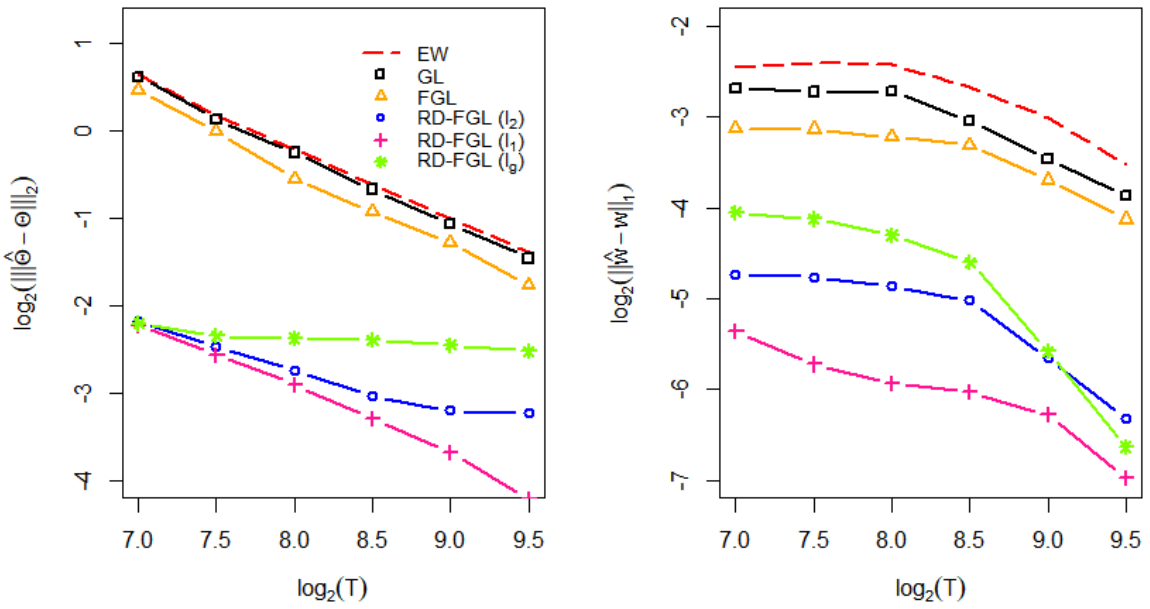

Figure E.8: **Averaged errors of the estimators of $\mathbf{\Theta}$ (left) and $\mathbf{w}$ on logarithmic scale (base 2): one break in $\mathbf{\Theta}_\varepsilon$.** $p = T^{0.85}$, $q = 2(\log(T))^{0.5}$, $s_T = \mathcal{O}(T^{0.05})$. The horizontal axis ranges from $T = 2^7, \ldots, T = 2^{9.5}$.

As demonstrated in Figure E.8, similarly to the findings in the main manuscript, accounting for the break significantly reduces the estimation error of precision matrix and combination weights even if the break magnitude is small.

Second, we explore behavior of MSFE for smaller break magnitude. The setup is the same as in Subsection E.1: the period for training the models is set to be $m_1 = m/3$ – this is used to train competing FAR models in (E.3). The remaining part of the sample,

$m_2 = m - m_1$ is split as follows: the estimation window for training competing models is set to be window $= m_2/2$. We roll the estimation window over the test sample of the size $m_2/2$. The break point is fixed at $1/2$ of the first estimation window. Before the break, when generating θ_s we set $c_2 = 0.3$, and after the break $c_2 = 0.4$. All other parameters stay unchanged.

Similarly to the main manuscript, we include different specifications of the smoothing function $\psi(\cdot)$. Figure E.9 shows the performance of all models including RD-FGL with γ estimated using cross-validation: similarly to the conclusions in Subsection E.2, accounting for the break significantly reduces MSFE of the combined forecast even if the break magnitude is small.

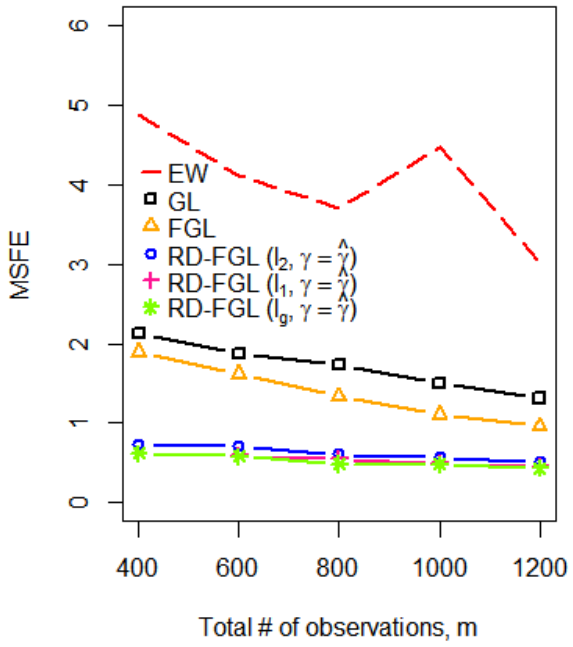

Figure E.9: **Plots of the MSFE over the total number of observations m.** $c_1 = 0.75$, $c_2 = 0.3$ (before the break), $c_2 = 0.4$ (after the break), $M = 100$, $r = 5, \sigma_\xi = 1$, $L = 7$, $K = 2$, $p = 24$, $q = 3$, $\rho = 0.9$, $\phi = 0.8$.

Appendix F Additional Figures

Figures F.1-F.2 provide additional illustration of the stylized fact that the ECB SPF respondents tend to jointly understate or overstate the predicted series.

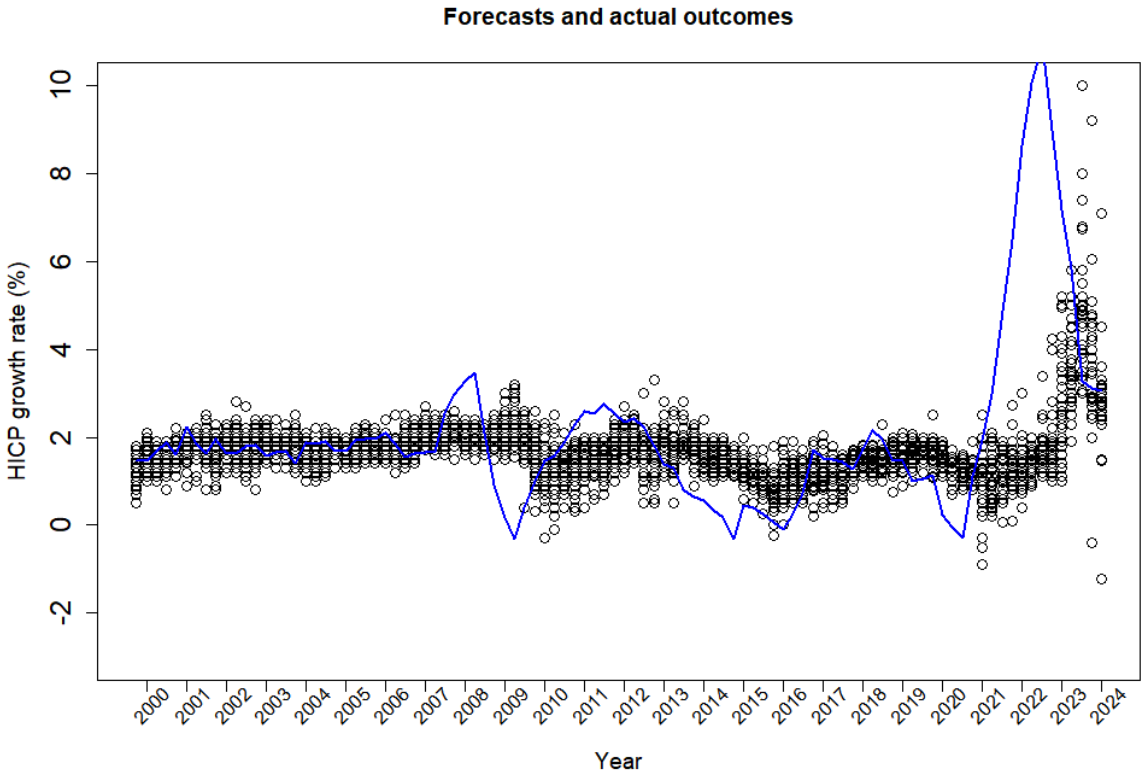

Figure F.1: **ECB SPF on Harmonised Index of Consumer Prices (HICP)**. Each circle denotes the forecast of each professional forecaster in the SPF for the quarterly 2-quarters-ahead forecasts of Euro-area inflation, year-on-year percentage change of the Harmonised Index of Consumer Prices. Actual series is the blue line. *Source: European Central Bank*.

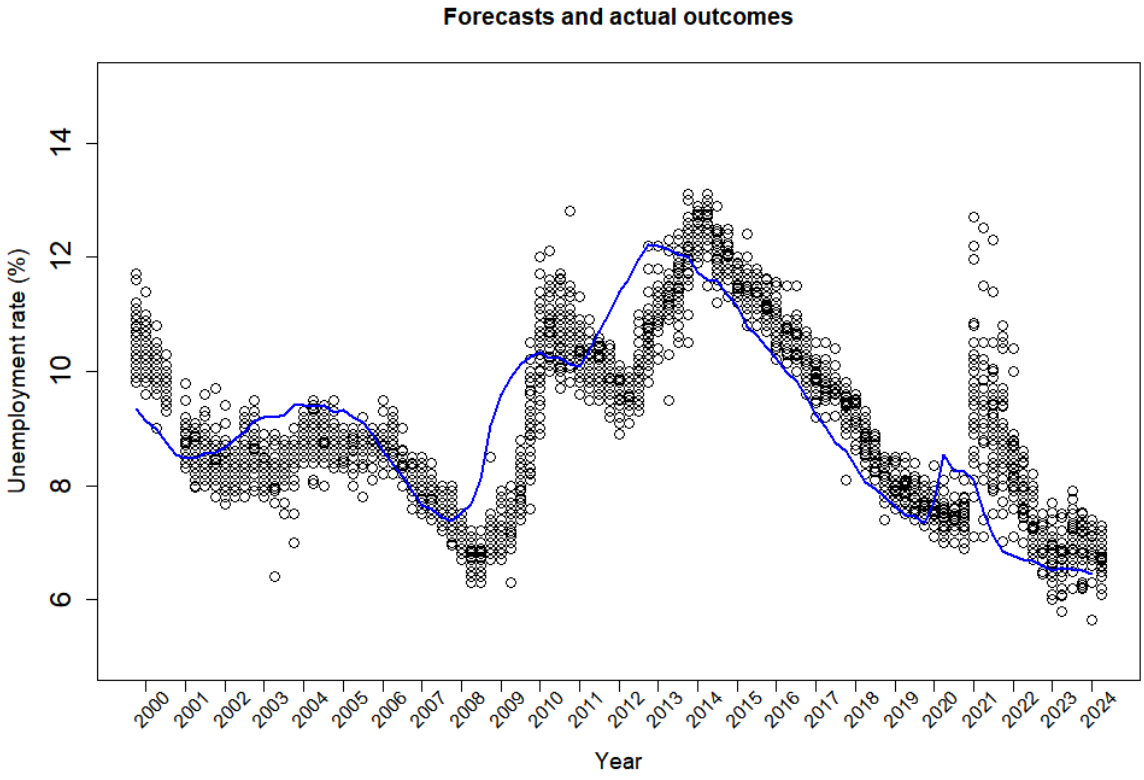

Figure F.2: **ECB SPF on Unemployment Rate**. Each circle denotes the forecast of each professional forecaster in the SPF for the quarterly 2-quarters-ahead forecasts of Euro-area unemployment rate, percentage of the labor force. Actual series is the blue line. *Source: European Central Bank*.

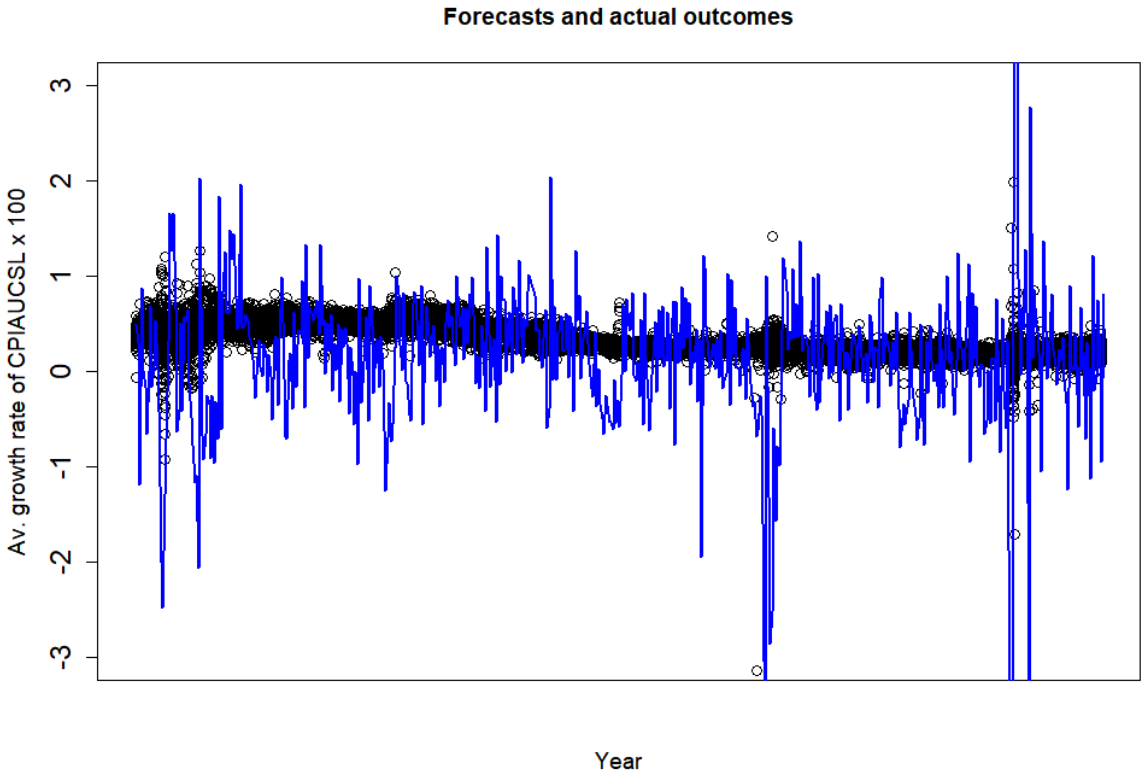

Figure F.3: **FRED-MD on consumer price index for all items (CPIAUCSL)**. Each circle denotes two-step ahead forecast of the average growth rate of CPIAUCSL times 100 based on individual monthly indicators (125 forecasting models per period). Actual series is the blue line.

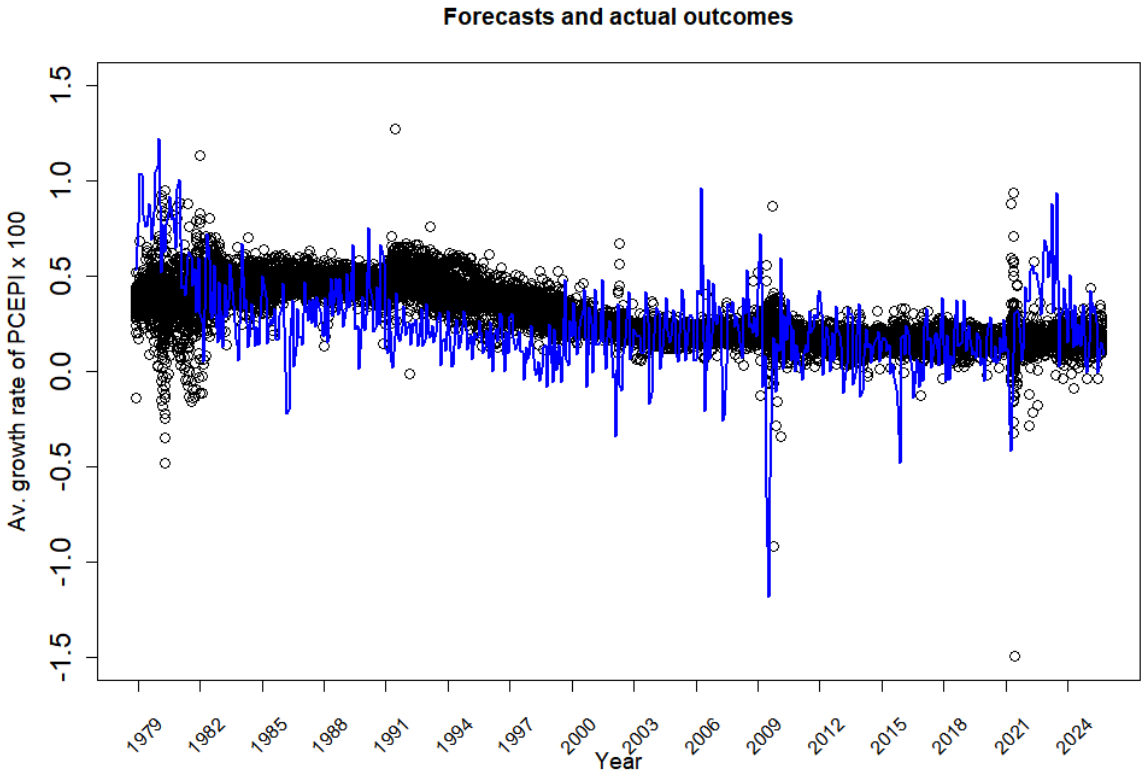

Figure F.4: **FRED-MD on Chain Index (PCEPI)**. Each circle denotes two-step ahead forecast of the average growth rate of personal consumer expenditures: PCEPI times 100 based on individual monthly indicators (125 forecasting models per period). Actual series is the blue line.

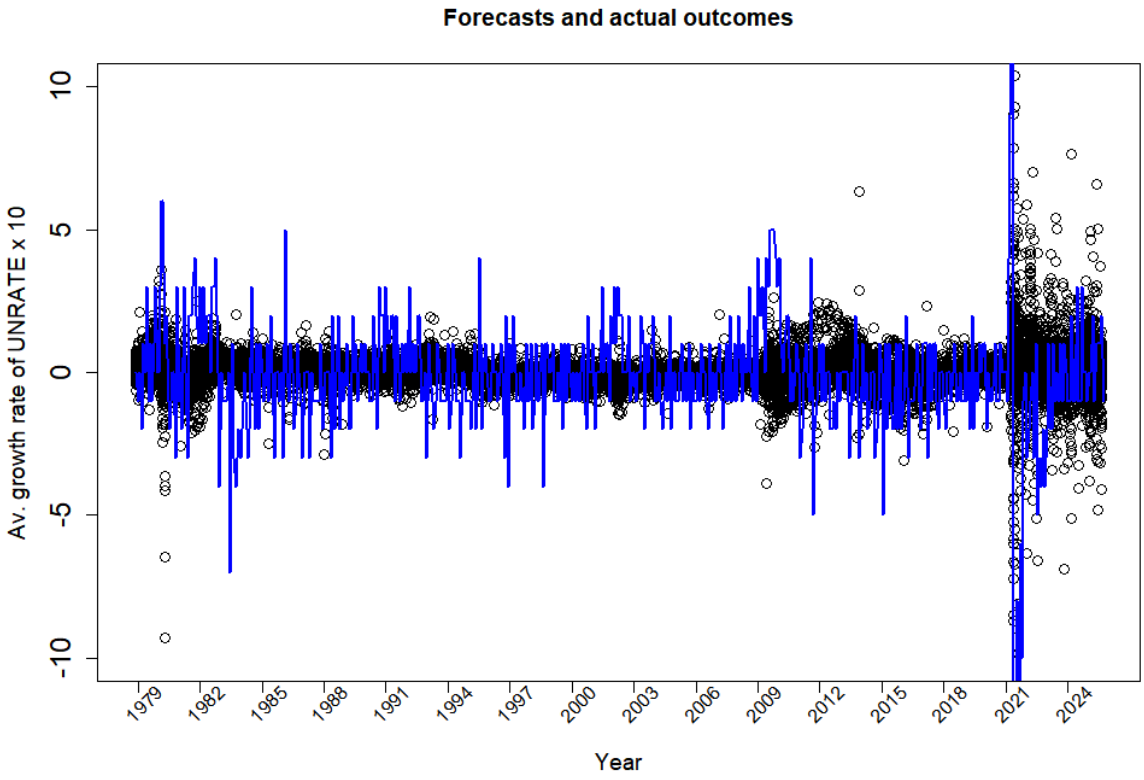

Figure F.5: **FRED-MD on civilian Unemployment Rate (UNRATE)**. Each circle denotes two-step ahead forecast of the average change of UNRATE times 10 based on individual monthly indicators (125 forecasting models per period). Actual series is the blue line.

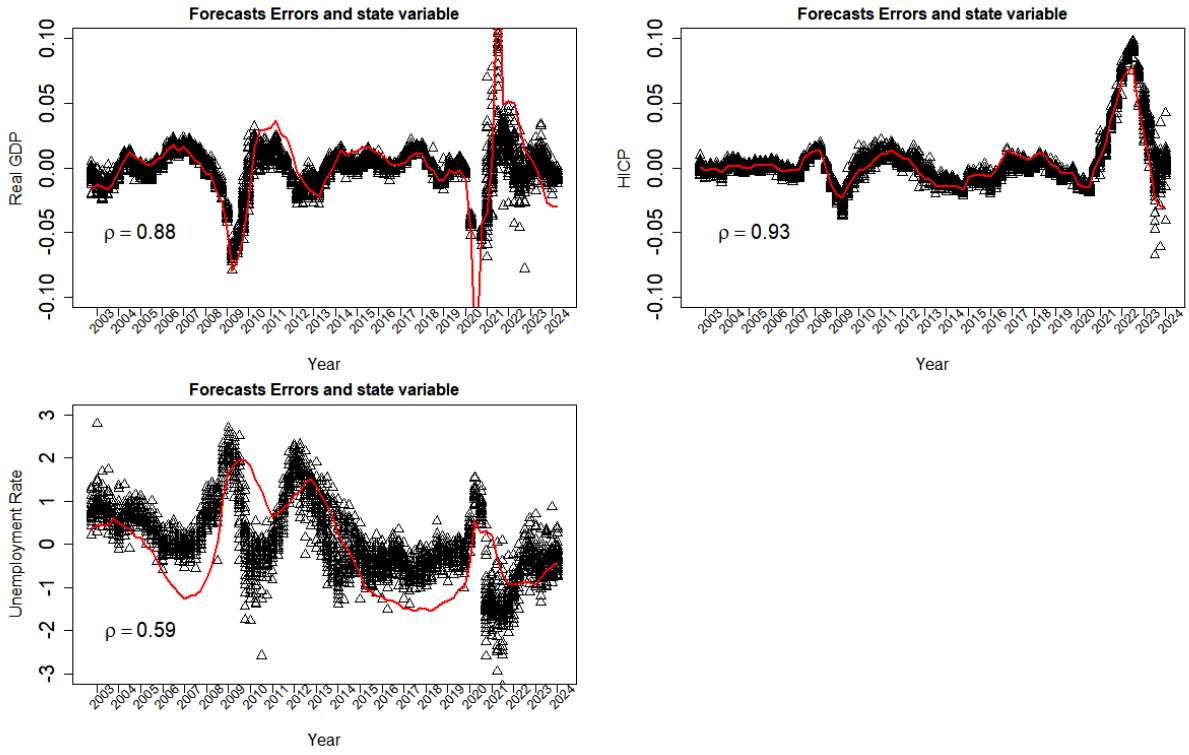

Figure F.6: **Forecast Errors and State Variable for ECB SPF**. Each triangle denotes a forecast error defined as the difference between the forecast and actual outcome (i.e., the difference between black circles and blue lines in Figures 1, F.1-F.2). Red line is a state variable defined as the difference between actual series and its four-quarters average. ρ is the correlation between the average value of forecast errors with the state variable.

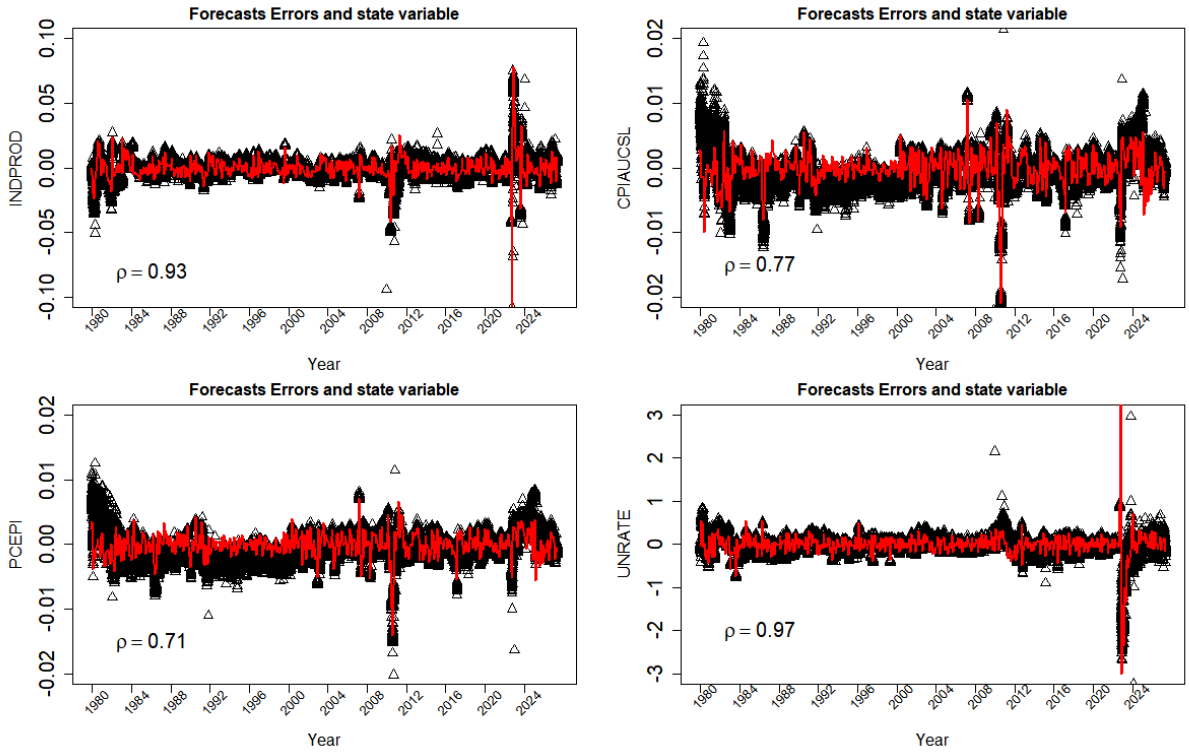

Figure F.7: **Forecast Errors and State Variable for FRED-MD**. Each triangle denotes a forecast error defined as the difference between the forecast and actual outcome (i.e., the difference between black circles and blue lines in Figures 2, F.3-F.5). Red line is a state variable defined as the difference between actual series and its twelve-months average. ρ is the correlation between the average value of forecast errors with the state variable.